PhD Dissertation

Università degli Studi di Trento

Dottorato di Ricerca in Matematica - XVI ciclo

Nicola Bertoldi

STATISTICAL MODELS AND SEARCH ALGORITHMS FOR MACHINE TRANSLATION

Advisors:
Dr. Marcello Federico
ITC-irst, Centro per la Ricerca Scientifica e Tecnologica

Prof. Roberto Battiti
Università degli Studi di Trento

February 2005

Acknowledgments

At this point, I would like to express my thanks to all people who supported and accompanies me during the progress of this work.

In particular, I would like to thank

Marcello Federico for his valuable advice, his continuous interest and numerous insightful discussions.

Prof. Roberto Battiti for kindly taking over the task of the co-referee for this work.

Prof. Dr-Ing. Hermann Ney and all people I met during my visit at the Lehrstuhl für Informatik VI in Aachen for their kind hospitality, especially Daniel, Evgeny, Nicola, and Richard.

My colleagues of the Hermes research line, Marcello, Mauro, Quan, Roldano, and Vanessa, for many constructive discussions and the very good working atmosphere.

All guys who had worked or still work at ITC-irst for after-work moments.

And all my friends who always had been close to me even when I had not.

The last and most grateful thank goes to all my family for the constant and tactful support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Machine Translation (MT) is the use of computer to automate the process of translating written or spoken texts from one language to another.
The problem of automatically producing high-quality translations of arbitrary texts is nowadays however hardly far from being solved. Nevertheless, currently available technology can be deployed to tackle less ambitious but still useful translation tasks. In particular, MT can be employed in information-acquisition tasks, for which a *rough translation* is adequate, in tasks where a *draft translation* can be improved by human post-editing, and in limited-domain tasks in which fully automatic high-quality translation is achievable. In general, MT becomes more difficult if the input is somehow corrupted. This is the case, for instance, when the input is supplied by a speech recognizer.
This thesis will focus on three specific translation tasks of increasing complexity, which will be introduced in Sections 1.1-1.3: *query translation*, *text translation* and *speech translation*.

1.1 Query Translation

Nowadays, with the enormous amount of multilingual information available on the World Wide Web, many Natural Language Processing applications, like Information Retrieval, Question Answering, and Text Classification, have began

to face the problem of crossing over the language barrier. For instance, Information Retrieval might consider searching for documents in languages different from that used to formulate the query. Similarly, in Question Answering one could be interested in finding the right answer to a given question, no matter about its language. Or, in Text Classification, one could need to cluster or classify documents of a multilingual collection. Recent literature shows that by exploiting MT techniques providing just rough translations the performance loss between monolingual and cross-lingual tasks is small. In this work, we concentrated on MT techniques suited to Cross-Language Information Retrieval.

1.1.1 Cross-Language Information Retrieval approaches

Cross-Language Information Retrieval (CLIR) can be approached by decoupling translation and retrieval tasks, and using existing MT and IR systems as "black boxes". But, as MT systems are far from being perfect, incorrect translations with meaning different from the original might negatively affect IR performance. As reported in [42], IR effectiveness can be improved when multiple alternative translations are used, but at the moment few MT systems provide them.

Moreover, many MT systems tentatively provide grammatically well-formed translations, which cannot be fully exploited by current IR systems. These are mainly based on the so-called *bag-of-words* models and just exploit *keywords*, or *content words*, of the texts they are working on. In fact, functional words, like articles, prepositions, modal verbs, pronouns are simply disregarded because they do not discriminate between documents, given that they are almost homogeneously spread within all texts. Hence, essential requirement of MT is the preservation of the meaning of the keywords across languages, rather than the production of well-formed translations.

An integrated approach between MT and IR is thus preferable, which possibly takes advantage of multiple translations of keywords. Alternative translations

can be easily obtained from bilingual dictionaries [31, 2, 64] and parallel corpora [54, 69]. Recent work on CLIR showed that simple statistical models for dictionary based translation outperform sophisticated MT models (e.g. Systran). Moreover, experiments reported in [27, 89] also suggest that bilingual dictionaries have to be complemented by a statistical language model to achieve good performance.

As full document translation in large corpora is quite costly and difficult, *query translation*, i.e. the translation of the query keywords, has been the major focus of research in the area of CLIR [2, 27, 28, 89]. In this work we follow the same strategy.

1.1.2 MT issues in Query Translation

In [30], main issues related to query translation have been identified. Whatever resource is employed in query translation, it has to provide good *coverage* of the source and target vocabularies. Names of entities, like people and locations, are frequently used in queries for news article and their translation is often not trivial; many geographical names have different spelling (Milano, Milan, Mailand) or even different roots (Deutschland, Germany, Allemagne) in different languages. Often acronyms of organizations are also different (UN, ONU, UNO). If CLIR is applied to languages with different alphabets, like Cyrillic, Arabic, or Chinese, we have to tackle out the problem of different transliterations of named entities (Jeltsin, Eltsine, Yeltsin, Jelzin).

The *sense disambiguation* of translation alternatives is a another major issue. While some translations can be appropriate for a query because they mostly preserve the original meaning, others should be discarded because they are completely wrong. Moreover, as term weighting is crucial for IR, acceptable translations should be weighted and ranked in accordance with their closeness to the original meaning of the query.

1.2 Text Translation

Undoubtedly, *Text Translation*[1], i.e. the translation of written texts, has been the main battlefield of MT research since the pioneer attempts [86]. Good overviews of the history and approaches to MT can be found in [35, 57].

1.2.1 MT issues in Text Translation

Although its long history, MT research has not yet achieved a level of quality to permit its widespread application. This apparent unsuccess is mainly due important differences existing between many human languages. Texts can be written with characters (Roman, Cyrillic, Arab alphabets) or ideograms (Japanese, Chinese); words can be composed by one (Vietnamese, Cantonese) or many morphemes (Eskimo); morphemes can be agglutinated (Turkish) or fused (Russian); verb, subject and object can be ordered as SVO (English, French), as SOV (Japaneses, Hindi), as VSO (Irish, Hebrew); constraints are given (English, Italian) or not (Mandarin) for gender and number of articles, nouns and pronouns; verbs are declined (Italian) or not (Mandarin) with respect to tense and mood; compound names are widely (German) or poorly (Italian) used.
Besides grammatical differences, other issues relate nouns and their meaning. As several words have multiple meanings and usually more translations (“wall” into “Mauer” or “Wand”), sense disambiguation is necessary. Furthermore, single words can be translated into several words (“informatica” into “computer science”) and viceversa (“per favore” into “please”). Sometimes, lexical gaps can be so strong that a concept expressed by a single word in a language can not be translated unless using a long sentence (Japanese “oyakoko” means “we make do with filial piety”).
Finally, an hypothetical perfect MT system should also consider the text genre; e.g. it would not be acceptable to translate a novel with scientific terms.

[1] With a little abuse of terminology, we will use the term MT to refer to Text Translation throughout the thesis.

All the above problems, and many philosophical issues, prove the intrinsic complexity of the MT task. However, by reducing expectations, MT systems can effectively be used for translating under controlled conditions, like, for instance, in limited domains, where word ambiguity is low and text genre is fixed.

1.2.2 Approaches to Machine Translation

Historically, most approaches to MT fall into one of the following three types: *interlingua*, *transfer*, and *direct*.

The *interlingua* approach is based on the assumption that the content of a task oriented text can be well approximated and automatically mapped into a relatively simple artificial canonical language, called interlingua[2]. Hence, content based translation from the interlingua to any language can be carried out by developing a suitable natural language generation modules. Main advantage of the interlingua approach is the decoupling of the translation into apparently simpler problems: the analysis of the source input for representing its meaning in the interlingua, and the synthesis of the output from the interlingua. Moreover, the same interlingua can be used as a pivot for many languages, reducing the effort of developing MT systems for other language pairs. The main drawback is the difficulty to develop an interlingua which is sufficiently complete and consistent, in order to cover all possible expressions in the domain, and which should be, at the same time, easy to generate and interpret automatically. For this reason, at the moment, the interlingua approach has been applied in very limited domains. Last but not least is the inevitable loss of information induced by the interlingua representation.

The *transfer* approach performs translation at the level of grammatical structures, by applying contrastive knowledge, i.e. knowledge about differences between languages. This approach tries to alter the syntactic structure of the input to conform it to the rules of the target language. In particular, the transfer ap-

[2]Defining an interlingua means also create an *ontology* of the task domain.

proach involves three phases: *analysis*, *transfer*, and *generation*. The input in the source language is first syntactically parsed into an abstract internal structure (usually a parse tree); thereafter, this structure is transferred into a corresponding structure in the target language; finally, the translation is generated. The lexical transfer is usually performed during the syntactic transfer for functional words and during the generation step for content words. The level of transfer can vary from purely syntactic deep structure markers to syntactic-semantic annotated trees. Transfer approach involves bilingual resources; hence, a different MT system should be tailored for each language pair.

On the opposite side of the interlingua strategy is the *direct* approach. It is based on the philosophy that MT system should perform as little deep analysis as possible. Words and syntax of the source input has to be analyzed only the strictly necessary to resolve ambiguities, to identify the appropriate target expressions, and to find the word order of the output. A direct MT system is typically composed of several modules, each focusing on a specific problem: morphological analysis, lexical transfer of content words, processing of functional words, identification and transfer of idioms, phrases, and compounds, syntactic and morphological processing of the target text, and word reordering. MT systems differ in the order and use of such modules. It is worth remarking that a direct MT system is designed for a specific language pair. A feature of the direct approach is that each problem is solved in one stage, including analysis, transfer, and generation aspects. Solving problems one at a time may be more tractable. Direct MT systems tend to be conservative, in the sense they only reorder words when required by obvious ungrammaticality in the target output. Perhaps the key characteristic of direct models is that they work without complex structures and representations, as the two previous approaches, which indeed require a deep knowledge about languages.

1.2.3 Rule-based versus empirical methods

The previous classification answers the questions of what representation to use and what steps to perform to translate. An orthogonal classification relates the methods used in the development of language processing. We distinguish between *rule-based* and *empirical* methods.
In the rule-based systems, experts specify a set of rules aiming at modeling the translation process. This approach is very expensive because it requires human work, to create well-defined rules which should cover all linguistic aspects of a language, exceptions included.
The empirical, or *data-driven*, approach instead acquire translation knowledge automatically from analysis of a large sample translations. The main advantage of such approach is that an MT system can be built very quickly for new language pairs and new domains, whenever a suitable amount of data is available. However, the amount of data needed to develop a system depends on the complexity of the domain. Empirical methods usually apply to transfer or direct MT approaches. Statistical MT systems instead exploits translation examples to define a statistical MT (SMT) model. Most SMT models are based on the source channel paradigm or Maximum Entropy framework. Example-based systems provide translation of a new sentence by analyzing previously seen translation example [76, 13].

1.3 Spoken Language Translation

The translation task becomes more difficult when the input to be processed is spoken language. In fact, an additional level of complexity is given by the necessity of correctly recognizing the content of the speech signal. Unfortunately, Automatic Speech Recognition (ASR) systems are far from being perfect; thus, the recognition step usually adds noise in the translation process. Moreover, spoken language is usually not syntactically well-formed and might contain

spontaneous speech phenomena, such as hesitations and repetitions. Transcription errors can corrupt either syntax or meaning of the utterance. While the former might be recovered during translation, the latter is hardly more difficult to handle. At the moment, SLT systems focus on limited domains, like air travel queries, appointment scheduling, hotel reservations, etc.

1.3.1 SLT approaches

Spoken Language Translation (SLT) systems feature an ASR module and a MT module. SLT systems differ in the level of integration between the two modules and the technology used for developing them. First approaches to SLT task simply concatenate the two module: the best transcription provided by the ASR system is fed to an MT system which produces the final translation. This cascade strategy does not add any further difficulty, but does not allow recovering from speech recognition errors.

More integration can be achieved either by supplying alternative transcription hypotheses to the MT system, or by developing a system which directly translates the speech utterance, without transcribing it before. In the former approach, the MT system usually process a a set of most probable transcriptions hypotheses provided by the ASR system. Recently, the exploitation of a *word graph* of hypotheses generated by the ASR systems has been considered.

Finite State Transducers are also used, because they provide a suitable framework to integrate ASR and MT decoders [17]. Unfortunately, a drawback of this approach is that it is hardly scalable to large domains.

Chapter 2

Scientific goals

This thesis aims at extending the state-of-the-art in three MT tasks namely *query translation*, *text translation*, and *spoken language translation*. As statistical approach competes very well, or even ouperforms, rule-based methods if suitable amount of data are available, and does not require any human expertise for developing MT systems, we focus on the this framework during our PhD research. Even if the statistical MT systems can apply to any language pair, in this thesis they are used to translate between Italian and English and from Chinese to English.

- Main approaches to Information Retrieval feature the well-known and very performing Okapi formula. However, applying this method to Cross-Language Information Retrieval is not straightforward. Instead, recent approaches based on statistical model provide a more suitable framework for including translation into IR. We propose an original statistical model, which tightly combines a model for translating keywords of the queries and a language model to score relevance between queries and documents. Interestingly, the query-translation model exploits co-occurrences of terms within the target collection and a bilingual dictionary.

- Recent work in statistical MT has shown that translation performance can be boosted by exploiting phrase-based translation models. Solutions have

been proposed which depart from the popular standard word-based models. We propose a new statistical translation model that extends in an easy way the hystorically leading Model 4 proposed by [12]. Moreover, by looking at the model as a log-linear model, a Minimum Error Training procedure can be applied to optimize system performance.

Furthermore, we implement a search algorithm based on Dynamic Programming, which directly derives from the generative process of the model. The algorithm, which is sinchronous with the output string, applies both to word- and phrase-based models.

Literature on Automatic Speech Recognition showed that rescoring multiple hypotheses improve overall performance. We think that this is true also for Machine Transaltion. On this way, we modify our search algorithm to output not just the best translation, but a word graph containing all hypotheses considered during the decoding. The availability of multiple hypotheses would permit to apply a rescoring strategy which exploits new additional knowledge difficult to embed into the original model.

- Spoken Language Translation is a very recent research field; hence, any effort to develop new methods and models can improve the state-of-the-art. Use of multiple transcription hypotheses has been shown to be effective in recovering transcription errors. At the moment, most approaches consider the ASR system and the MT system as two separate modules, and simply fed up the MT decoder with a list of the N-best transcriptions. The disadvantage of these methods is that we have to run the MT decoder N times. Our idea is translating all alternatives in one time. We propose a more integrated approach to combine acoustic features and phrase-based translation model. A statistical translation model is defined, which apply directly to the word-graph generated by the ASR system, slightly modified for computational reasons. The search algorithm developed for text translation can

be used for translating the modified word graphs, apart from some minor changes.

Our main expectation is to significantly reduce computational effort in terms of both decoding time and memory consumption and, obviously, achieve comparable performance.

Chapter 3

Query Translation

Our first approach to MT focused on the *keyword* translation, i.e. the translation of content words of a text. Keyword translation is usually not a research field per se, but rather it is studied to cross over the language barrier in other NLP applications, like Information Retrieval, Text Classification, and Question Answering. These applications achieve their goal mainly exploiting content words of documents or queries; when more languages are involved, a translation module has to provide an effective way to correctly transfer the meaning of such words from one language to another.

We only concentrated on Cross-Language Information Retrieval, and, in particular, on the *query translation*. In fact, following a popular trend, we apply keyword translation to queries only, because a full document translation in large corpus is quite costly.

Information Retrieval (IR) is the task of finding documents, inside a known collection, which are relevant to a given topic or query. If topics and documents are written in the same language, e.g. English, we have so called monolingual IR, otherwise Cross-Language IR (CLIR) occurs. In particular, if only two languages are involved, e.g. French for queries and English for documents, IR is called bilingual; if the collection contains documents in more than one language, e.g. English, French and Italian, IR is instead called multilingual. As

habit in the IR literature and with no risk of confusion, henceforth, the term CLIR will be used only to mean bilingual IR.

This Chapter presents the IR system developed at ITC-irst to tackle monolingual and bilingual IR. After a brief overview of main approaches to CLIR, given in Section 3.1, the monolingual IR system is presented in Section 3.2. The system features three different models for matching topics against documents: a statistical language model, an Okapi model, and a combination of the two approaches. Section 3.3 presents the statistical query-translation model used for CLIR. Training data for the translation model consists in a bilingual dictionary and the target document collection. Section 3.5 introduces the bilingual IR system featuring a statistical framework which couples two basic components: a query translation model, based on *hidden Markov models* [68], and a retrieval model which works as in the monolingual case. The two models can be either put in cascade or tightly coupled. The latter case results in a probability score computed by integrating over a set of possible translations of the query. Performance of the system is discussed in Section 3.7 for two particular tasks: Italian monolingual retrieval and the Italian-English biolingual retrieval at the Cross-Language Evaluation Forum (CLEF) from 2000-2004. Section 7.1 concludes this Chapter with a discussion about interesting issues which emerged from our research in this area..

3.1 Previous work

For what concerns IR, two main approaches can be identified in the literature. The former ranks documents by weighting every term in the query according to its relevance within each document and the whole collection. Okapi [70] is the name of a retrieval system project that developed a family of such scoring functions.

Alternatively, matching between query and documents is computed by means

of probability distributions estimated on the target collection. Different statistical language modeling techniques were proposed [49, 53] to combine relative frequencies of each document with those of the whole collection.

Recent work on CLIR has shown that simple statistical models for dictionary-based translation outperform more sophisticated machine translation models (e.g. Systran). Many research groups proposed statistical translation models based on available dictionaries [43, 23] or automatically trained from parallel corpora [54]. A back-off translation model was proposed in [69], which combines evidence from dictionary-based and corpus-based statistics. In dictionary-based approaches, alternative translations are used for query expansion [34, 72], however no context is considered for the sake of disambiguation.

Improvement were proposed by considering word co-occurrence statistics within the target collection. In [27], translations of query terms are selected which co-occur most often with all other alternatives in the target documents. This strategy is improved by adding decaying factors which reduce the effect of co-occurrences as term distance increases, following the intuition that closer terms are more strongly correlated [28].

Probabilistic CLIR systems have been developed, which use generative models to estimate the probability that a target document is relevant for a given query. Systems based on the probabilistic framework mainly differ in the decomposition of the basic distribution and on the dependency assumptions among query terms. In particular, [31, 89] assume independence between the query terms, and do not use Markovian assumptions in the translation process.

Standard statistical machine translation models presented in [12] were applied to IR by [3].

q, f, e	generic term, term in French, term in English
$\mathbf{q}$, $\mathbf{f}$, $\mathbf{e}$	generic query, query in French, query in English
$\mathcal{D}$, d	collection of documents, generic document
$\mathcal{V}$, $\mathcal{V}(d)$	number of different terms in $\mathcal{D}$, and in document d
N, $N(d)$	number of term occurrences in $\mathcal{D}$, and in document d
$N(q)$, $N(d,q)$, $N(\mathbf{q},q)$	frequency of term q in $\mathcal{D}$, in document d, and in query $\mathbf{q}$
N_q	number of documents in $\mathcal{D}$ which contain term q
$\bar{l}$	average length of documents in $\mathcal{D}$

Table 3.1: List of often used symbols.

3.2 Monolingual IR

Formally, monolingual IR can be approached as follows: given a query $\mathbf{q} = q_1, \ldots, q_n$, rank all documents d in a collection $\mathcal{D}$ according to a probability or a scoring function $\mathcal{S}(\mathbf{q}, d)$, which measures the relevance of d with respect to $\mathbf{q}$. In the following, three query-document matching criteria are introduced. The first is based on a statistical language model (LM), the second is derived from the Okapi framework, and the last is a combination of the first two. Main notation used in the following is summarized in Table 3.1.

3.2.1 Language model

The relevance of a document d with respect to a query $\mathbf{q}$ can be expressed through a joint probability, which can be decomposed as follows:

$$\Pr(\mathbf{q}, d) = \Pr(\mathbf{q} \mid d) \Pr(d) \tag{3.1}$$

where $\Pr(\mathbf{q} \mid d)$ represents the likelihood of $\mathbf{q}$ given d, and $\Pr(d)$ represents the a-priori probability of d. By assuming no a-priori knowledge about the documents and an order-free multinomial model for the likelihood, the following

probability score can be derived:

$$\Pr(\mathbf{q}, d) \propto \prod_{i=1}^{n} \Pr(q_i \mid d) \tag{3.2}$$

By taking the logarithm, we can define the following scoring function:

$$lm(\mathbf{q}, d) = \sum_{q \in \mathbf{q}} N(\mathbf{q}, q) \ \log \Pr(q \mid d) \tag{3.3}$$

where the sum is over the set of terms in the query $\mathbf{q}$.

The probability $\Pr(q \mid d)$ that a term q is generated by d can be estimated by applying statistical language modeling techniques [25]. Previous work [49, 53] proposed to interpolate relative frequencies of each document with those of the whole collection, with interpolation weights estimated by maximum likelihood on the documents. Here, the same interpolation scheme is applied but weights are estimated according to the smoothing method by [87]. In particular, word frequencies of a document are smoothed linearly and the amount of probability assigned to never observed terms is made proportional to the number of different words contained in the document. Hence, the following probability estimate results:

$$\Pr(q \mid d) = \frac{N(d,q)}{N(d) + \mathcal{V}(d)} + \frac{\mathcal{V}(d)}{N(d) + \mathcal{V}(d)} \Pr(q) \tag{3.4}$$

where $\Pr(q)$, the word probability over the collection, is estimated by interpolating the smoothed relative frequency with the uniform distribution over the collection's vocabulary V:

$$\Pr(q) = \frac{N(q)}{N + \mathcal{V}} + \frac{\mathcal{V}}{N + \mathcal{V}} \frac{1}{\mathcal{V}}. \tag{3.5}$$

3.2.2 Okapi model

Okapi [70] is the name of a retrieval system project that developed a family of scoring functions. According to the Okapi framework, every term in the query

is weighted according to its relevance within a document and within the whole collection. In our IR system the following function was used:

$$okapi(\mathbf{q},d) = \sum_{q \in \mathbf{q}} N(\mathbf{q},q)\, W_d(q) \log W_{\mathcal{D}}(q) \tag{3.6}$$

where:

$$W_d(q) = \frac{N(d,q)(k_1+1)}{k_1(1-b)+k_1 b\frac{N(d)}{\bar{l}}+N(d,q)} \tag{3.7}$$

weighs the relevance of the term q inside the document d, and:

$$W_{\mathcal{D}}(q) = \frac{N-N_q+0.5}{N_q+0.5} \tag{3.8}$$

is the term inverted document frequency, which weighs the relevance of term q inside the whole collection $\mathcal{D}$.

Parameter values $k_1 = 1.5$ and $b = 0.4$ were empirically estimated [7] on some development data. It is worth noticing that our scoring function corresponds to the well known BM25(k_1,k_2,k_3,b) model [70], with the setting $k_2 = 0$, $k_3 = \infty$, $k_1 = 1.5$ and $b = 0.4$.

The Okapi and the language model scoring functions present some analogy. In particular, Equation (3.6) can be put in a probabilistic form which maintains the original ranking, thanks to the monotonicity of the exponential function. Hence, a joint probability distribution can be defined which, disregarding a normalization constant factor, is:

$$\Pr(\mathbf{q},d) \propto \prod_{i=1}^{n} W_{\mathcal{D}}(q_i)^{W_d(q_i)} \tag{3.9}$$

Henceforth, query-document relevance models will be indicated by the joint probability $\Pr(\mathbf{q},d)$, regardless of the used model, unless differently specified.

3.2.3 Combined method

By looking at the Italian monolingual runs of our first participation in CLEF 2000 [7], it emerged that the LM and the Okapi model have quite different be-

haviors. This suggested that if the two methods rank documents independently, more information about the relevant documents could be gained by integrating the scores of the two methods.

In order to compare the rankings of two models, the Spearman's rank correlation [50] was applied, which confirmed some degree of independence between the two information retrieval models. Hence, a combination of the two models [7] was implemented by just taking the sum of scoring functions, namely $lm(\mathbf{q},d)$ and $okapi(\mathbf{q},d)$. Actually, in order to adjust scale differences, single scores were re-scaled in the range $[0,1]$ before summation. Normalization was computed over union of the 300 top ranking documents of each method. It can be shown that summation of the normalized scores corresponds to a multiplication of probabilities, according to the above defined joint probabilities.

3.3 Query translation model

Query translation for CLIR is based on a hidden Markov model (HMM) [68], in which the observable part is the query $\mathbf{f}$ in the source language, e.g. French, and the hidden part is a corresponding query $\mathbf{e}$ in the target language, e.g. English. The model only assumes that the two queries have the same length. The joint probability of a pair $(\mathbf{f},\mathbf{e})$ is computed as follows:

$$\Pr(\mathbf{f}=f_1,\ldots,f_n,\mathbf{e}=e_1,\ldots,e_n)=\prod_{k=1}^{n}\Pr(f_k \mid e_k)\Pr(e_k \mid e_{k-1}) \tag{3.10}$$

Equation (3.10) puts in evidence two different conditional probabilities: the term translation probabilities $p(f \mid e)$ and the target LM probabilities $p(e \mid e')$. Probabilities $Pr(f \mid e)$ are estimated from a translation dictionary as follows:

$$\Pr(f \mid e)=\frac{\delta(f,e)}{\sum_{f'}\delta(f',e)} \tag{3.11}$$

where $\delta(f,e)=1$ if the English term e is one of the translations of the French term f and $\delta(f,e)=0$ otherwise.

Probabilities $\Pr(e \mid e')$ are estimated on the target document collection, through the following bigram LM, that tries to compensate for different word orderings induced by the source and target languages:

$$\Pr(e \mid e') = \frac{\Pr(e, e')}{\sum_{e''} \Pr(e, e'')} \tag{3.12}$$

where $\Pr(e, e')$ is the probability of e co-occurring with e', regardless of the order, within a text window of fixed size. Smoothing of the probability is performed through absolute discounting and interpolation [25] as follows:

$$\Pr(e, e') = \max\left\{\frac{C(e, e') - \beta}{N}, 0\right\} + \beta \Pr(e) \Pr(e') \tag{3.13}$$

$C(e, e')$ is the number of co-occurrences appearing in the corpus, $\Pr(e)$ is estimated according to Equation (3.5), and the absolute discounting term β is equal to the estimate proposed in [52]:

$$\beta = \frac{n_1}{n_1 + 2n_2} \tag{3.14}$$

with n_k representing the number of term pairs occurring exactly k times in the corpus.

3.4 Search algorithm

Generation of N-best translations with the proposed model can be efficiently performed with a simplified version of the tree-trellis based search algorithm by [74]. Briefly, the algorithm is based on two steps: a Viterbi search [68] proceeding forward along the source query, and an A^* search algorithm [56] proceeding backwards.

The Viterbi search algorithm (Table 3.4), computes the optimal translation of an input query $\mathbf{f} = f_1, \ldots, f_n$. The algorithm uses dynamic programming to compute, for each position t along $\mathbf{f}$ and translation e of f_t, the best translation

up to t and ending in e. This is carried out by finding the optimal continuation of all (optimal) translations computed at time $t-1$ (step 6). At each stage t, a backward link to the best incoming translation at time $t-1$ is set (step 7). Finally, the optimal complete translation is obtained by following the backward chain starting from to the best translation at time n. It is easy to see that the complexity of this algorithm is, in the average case, $O\left(n\bar{\mathcal{E}}^2\right)$, assuming $\bar{\mathcal{E}}$ is the average number of translations for a term.

1.	Input $\mathbf{f} = f_1, \ldots, f_n$
2.	Initialize for all $e \in \mathcal{E}(f_1)$ $Q[1,e] = P[f_1 \mid e]P[e]$
3.	Initialize for all $e \in \mathcal{E}(f_1)$ $B[1,e] = \varepsilon$
4.	For $t = 1, \ldots, n-1$
5.	for all $e \in \mathcal{E}(f_{t+1})$
6.	$Q[t+1,e] = P[f_{t+1} \mid e] \max_{e' \in \mathcal{E}(f_t)} Q[t,e']P[e \mid e']$
7.	$B[t+1,e] = \arg\max_{e' \in \mathcal{E}(f_t)} Q[t,e']P[e \mid e']$
8.	Backtrack solution $\mathbf{e}^*$:
9.	$e_n^* = \arg\max_{e \in \mathcal{E}(f_n)} Q[n,e]$
10.	$e_t^* = B[e_{t+1}^*]$ for $t = n-1, n-2, \ldots, 1$

Table 3.2: Viterbi search algorithm for query translation.

After a call to the Viterbi search, an A^* search (Table 3.4) is performed backward along the source query $\mathbf{f}$. At each iteration (step 9) it pops and examines the best partial theory (translation) from the stack *OpenSet*. If the translation covers the whole input query, then it is added to the N-best list (steps 10-12). Otherwise, all possible one-word expansions of it are computed (step 15). Scores are assigned to each expansion by combining the score g, computed by the translation model from the end to the current position (step 16), and the prediction score h computed by the Viterbi search which corresponds to the optimal translation from the start to the current position (step 17). Each theory expansion is then inserted into *OpenSet* so that it results ordered according to $g \cdot h$ (step 18).
The complexity of the A^* search algorithm is determined by the number of

1.	Perform Viterbi algorithm steps 1-7
2.	Initialize OpenSet=[],BestList=[], NB=0
3.	for all $e \in \mathcal{E}(f_n)$
4.	path=$[e]$
5.	h=$Q[n,e]$
6.	g=1
7.	insert (path,n,g,h) in OpenSet
8.	while OpenSet is not empty and $NB < N$
9.	$([e_n,\ldots,e_t],t,h,g) =$ pop OpenSet
10.	if $t = 1$
11.	append $[e_t,\ldots,e_t]$ to BestList
12.	$NB = NB + 1$
13.	else
14.	for all $e \in \mathcal{E}(f_{t-1})$
15.	path=$[e_n,\ldots,e_t,e]$
16.	g'=g $*P[e_t \mid e]P[f_{t-1} \mid e]$)
17.	h=$Q[t-1,e]$
18.	insert (path,t-1,h,g') in OpenSet

Table 3.3: Tree-trellis algorithm for the extraction of N-best query translations for CLIR application.

iterations times the cost of theory insertions (steps 6 and 16) performed at each step. As a theory at position i in the stack cannot expand into theories with a better score, each insertion operation just involves examination of the top N positions in $OpenSet$. Hence, at each iteration, the complexity of the insertion operations is $O(\bar{\mathcal{E}}N)$. The number of iterations is n to find the best translation and $O(n)$ to find the others. Hence, the total complexity of the A^* search is $O(n\,N^2\,\bar{\mathcal{E}})$.

3.5 Bilingual IR

From a statistical perspective, bilingual IR can be formulated as follows. Given a query $\mathbf{f}$, in the source language, one would like measure the relevance of a documents d, in the target language, by a joint probability $\Pr(\mathbf{f}, d)$. To fill the gap of language between query and documents, the hidden variable $\mathbf{e}$ is introduced, which represents a term-by-term translation of $\mathbf{f}$ in the target language. Hence, the following decomposition is derived:

$$\begin{aligned} \Pr(\mathbf{f}, d) &= \sum_{\mathbf{e}} \Pr(\mathbf{f}, \mathbf{e}, d) \\ &\approx \sum_{\mathbf{e}} \Pr(\mathbf{f}, \mathbf{e}) \Pr(d \mid \mathbf{e}) \\ &= \sum_{\mathbf{e}} \Pr(\mathbf{f}, \mathbf{e}) \frac{\Pr(\mathbf{e}, d)}{\sum_{d'} \Pr(\mathbf{e}, d')} \end{aligned} \tag{3.15}$$

In deriving Equation (3.15), one makes the reasonable assumption (or approximation) that the probability of document d given query $\mathbf{f}$ and translation $\mathbf{e}$, does not depend on $\mathbf{f}$. Equation (3.15) contains probabilities $\Pr(\mathbf{e}, d)$ and $\Pr(\mathbf{f}, \mathbf{e})$, which correspond, respectively, to the query-document and query-translation models described in the previous Sections.

In principle, the probability $\Pr(\mathbf{f}, d)$ results very expensive to compute. In fact, the main summation in (3.15) is taken over the set of possible translations of $\mathbf{f}$. As terms of $\mathbf{f}$ may typically admit more than one translation, the size of this set can grow exponentially with the length of $\mathbf{f}$. For instance, the Italian-English dictionary, used for our experiments, returns on average 1.84 English words for each Italian entry. Hence, the number of possible translations for a 40 word long query is in the order of 10^{10}! Finally, the denominator in Equation (3.15) requires summing over all document in $\mathcal{D}$ and should be computed for every possible translation $\mathbf{e}$.

The derivation of Equation (3.15) is of course not unique. Other types of statistical models for CLIR have been derived in the literature [31, 89, 3]. A comparative discussion of these models with respect to the one presented here can be found in [24]. Non statistical models for CLIR, dealing with multiple translations, are instead discussed in [64, 2].
Now, two algorithms are introduced which approximate, with increasing accuracy, the computation of Equation (3.15).

3.5.1 Cascade approach

A method to cope with the complexity of (3.15), is to apply the following maximum approximation:

$$\begin{aligned}
\Pr(\mathbf{f},d) &= \sum_{\mathbf{e}} \Pr(\mathbf{f},\mathbf{e}) \frac{\Pr(\mathbf{e},d)}{\sum_{d'} \Pr(\mathbf{e},d')} \\
&\approx \max_{\mathbf{e}} \left\{ \Pr(\mathbf{f},\mathbf{e}) \frac{\Pr(\mathbf{e},d)}{\sum_{d'} \Pr(\mathbf{e},d')} \right\} \\
&\approx \Pr(\mathbf{f},\mathbf{e}^*) \frac{\Pr(\mathbf{e}^*,d)}{\sum_{d'} \Pr(\mathbf{e}^*,d')} \\
&\propto \Pr(\mathbf{f},\mathbf{e}^*)\ \Pr(\mathbf{e}^*,d) \qquad (3.16)
\end{aligned}$$

where

$$\mathbf{e}^* = \arg\max_{\mathbf{e}} \Pr(\mathbf{f},\mathbf{e}) \qquad (3.17)$$

This approximation permits to decouple the translation and retrieval phases. Given a query $\mathbf{f}$, the Viterbi decoding algorithm is applied to compute the most probable translation $\mathbf{e}^*$, as explained in Section 3.3. Then, the document collection is searched by applying any monolingual IR model, explained in Sec-

tion 3.2, with the query $\mathbf{e}^*$. Table 3.4 shows the algorithm for the cascade approach.

1.	Input $\mathbf{f}$
2.	Compute the best translation of $\mathbf{f}$: $\mathbf{e}^* = \arg\max_{\mathbf{e}} Pr[\mathbf{f}, \mathbf{e}[$
3.	Order documents according to $P[\mathbf{e}^*, d]$

Table 3.4: Algorithm of the CLIR cascade approach.

3.5.2 Integrated approach

A more refined algorithm is now presented that relies on two approximations in order to limit the set of possible translations and documents to be taken into account in Equation (3.15).

Approximation 1. The first approximation redefines the query-translation probability by limiting its support set to just the N-best translations of $\mathbf{f}$, indicated by $\mathcal{T}_N(\mathbf{f})$. Hence,

$$\Pr'(\mathbf{f},\mathbf{e}) = \begin{cases} \dfrac{\Pr(\mathbf{f},\mathbf{e})}{K_1(\mathbf{f})} & \text{if } \mathbf{e} \in \mathcal{T}_N(\mathbf{f}) \\ 0 & \text{otherwise} \end{cases} \tag{3.18}$$

$K_1(\mathbf{f})$ is a normalization term which can be disregarded in Equation (3.15) for the sake of document ordering, as being constant with respect to the ranking variable d.

Approximation 2. A second approximation is introduced to reduce the computational burden of the denominator on Equation (3.15). Hence, the support set of the query-document model is limited to only documents which contain at least

1.	Input $\mathbf{f}$
2.	Compute $\mathcal{T}_N(\mathbf{f})$ with scores $P[\mathbf{f},\mathbf{e}]$
3.	For each $\mathbf{e} \in \mathcal{T}_N(\mathbf{f})$
4.	$N = 0$
5.	For each $d \in \mathcal{I}(\mathbf{e})$
6.	Compute $P[\mathbf{e},d]$
7.	Update $N = N + P[\mathbf{e},d]$
8.	For each $d \in \mathcal{I}(\mathbf{e})$
9.	Update $P[\mathbf{f},d] = P[\mathbf{f},d] + P[\mathbf{e},d] * P[\mathbf{f},\mathbf{e}]/N$
10.	Order documents according to $P[\mathbf{f},d]$

Table 3.5: Algorithm of the CLIR integrated approach.

one term of the query. Given a translation $\mathbf{e}$, let $\mathcal{I}(\mathbf{e})$ indicate the set of documents containing terms of $\mathbf{e}$. This set is easy to compute when the collection is accessed through an inverted index [26]. Hence,

$$\Pr'(\mathbf{e},d) = \begin{cases} \dfrac{\Pr(\mathbf{e},d)}{K_2(\mathbf{e})} & \text{if } d \in \mathcal{I}(\mathbf{e}) \\ 0 & \text{otherwise} \end{cases} \tag{3.19}$$

where $K_2(\mathbf{e})$ is a normalization term that occurs both in the numerator and denominator of the fraction in (3.15), and is therefore deleted. Thanks to this approximation, computation of the denominator in Equation (3.15) can be performed by summing up the scores of just the documents accessed through the inverted index.

The CLIR algorithm applying the two approximations is shown in Table 3.5. Briefly, given an input query $\mathbf{f}$, the N-best translations are computed first. Then, for each translation $\mathbf{e}$, the addenda in Equation (3.15) are computed only for documents containing at least one term of $\mathbf{e}$. This requires one additional loop over the documents in order to compute the normalization term. The complexity of the algorithm can be estimated as follows:

- $O\left(n\,\bar{\mathcal{E}}^2 + n\,N^2\,\bar{\mathcal{E}}\right)$ for step 2 in the average case [24]

- $o(n\,N\,\bar{I})$ for steps 3-9 in the average case
- $o(n\,\bar{\mathcal{E}}\,\bar{I}\,\log(n\,\bar{\mathcal{E}}\,\bar{I}))$ for step 10, in the worst case, i.e. N-best translations use all the available terms,

where n denotes the length of the query, N the number of generated translations, $\bar{\mathcal{E}}$ is the average number of translations of a term, and $\bar{I}$ is the average number of documents spanned by the inverted file index. The latter number is somehow controlled by the stop-term removal phase applied during document indexing. Generally, terms occurring in many documents are not considered significant for IR and are removed from the index. For instance, in the performed Italian-English experiments we had $\bar{I} \approx 110$ with $|\mathcal{D}| = 110,282$, and $\bar{\mathcal{E}}$=1.84.

Remark. It is worth noticing that the cascade approach is a special case of the integrated approach, which results by taking $N = 1$. The cascade method permits indeed to eliminate the normalization term in Equation (3.15).

3.6 The ITC-irst Cross-Language Information Retrieval system

The ITC-irst CLIR architecture is depicted in Figure 3.1. A query is first preprocessed as explained in Section 3.6.1, and then one of the two CLIR algorithm presented in Sections 3.5.1 and 3.5.2 is applied to retrieve the most relevant documents. Preprocessing is performed also on the document of the target collection. The required data for model training are also shown in the Figure.

3.6.1 Document and query preprocessing

Preprocessing of queries and documents aims at extracting keywords and removing functional terms. Some normalization is also performed to create equivalence classes of terms with the same root; this helps to reduce data sparseness

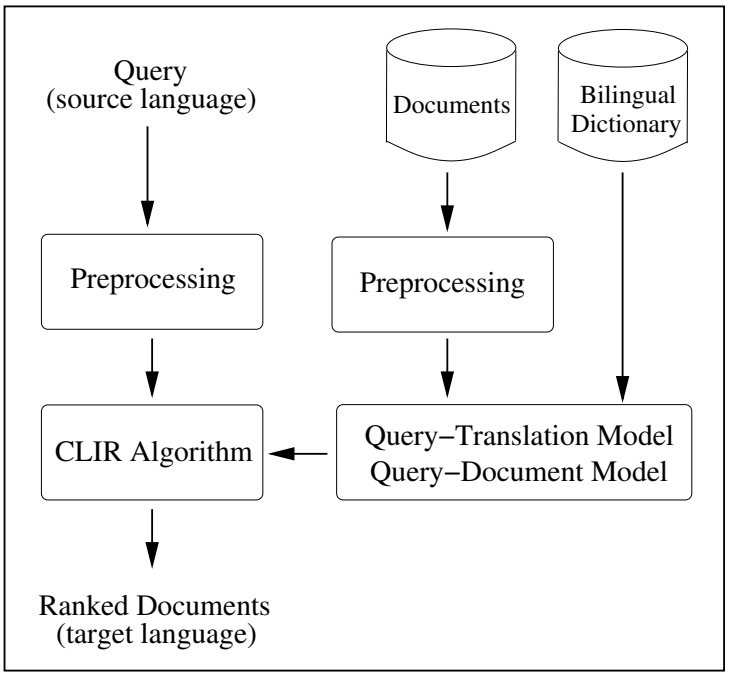

Figure 3.1: The ITC-irst Cross Language Information Retrieval system architecture.

and improve more robustness. A brief description of the modules used to preprocess Italian and English documents and queries is given in below. Tables 3.6 and 3.7 show, respectively, an original English topic and the its modifications through various preprocessing steps.

Tokenization Words are isolated from punctuation marks, abbreviations and acronyms are recognized, possible word splits across lines are corrected, and accents are distinguished from quotation marks.

Morpho-syntactic analysis Base forms of Italian words are obtained by combining morpho-syntactic analysis and statistical parts-of-speech tagging [5].

Stemming Word stemming is performed on English texts by using the Porter algorithm [65].

```
<top>
<num> C044 </num>
<EN-title> Indurain Wins Tour </EN-title>
<EN-desc> Reactions to the fourth Tour de France won by Miguel
Indurain. </EN-desc>
<EN-narr> Relevant documents comment on the reactions to the fourth
consecutive victory of Miguel Indurain in the Tour de France. Also
relevant are documents discussing the importance of Indurain in
world cycling after this victory. </EN-narr>
</top>
```

Table 3.6: English topic 44.

Stop-term removal Non relevant words are filtered out on the basis of their POS (only for Italian) and their inverted document frequency.

Handling multi-words When translating, phrasal verbs, noun phrases, compounds, have to be recognized and correctly transferred to the target language. As dictionaries typically contain many multi-word entries, these were included in the statistical translation model. Besides including them into the lexicon probabilities, co-occurrences of multi-words were also collected in the target LM. Multi-words were indeed not considered for the sake of indexing and retrieval. Hence, after the translation step they were split into single words.

Handling out-of-dictionary words As the query-translation model relies on a bilingual dictionary, an high coverage of the source and target languages is essentail to achieve good retrieval peformance. Dictionary coverage is artificially augmented by applying proper name recognition on the original query and by forcing verbatim translation of proper names (with English version stemmed) which do not occur in the dictionary. However, for proper names, guarantee about the correctness of the translation is lost; in fact, names of people are usually written with the same transliteration both

a	Title	Indurain Wins Tour.
	Desc.	Reactions to the fourth Tour de France won by Miguel Indurain.
b	Title	indurain win tour .
	Desc.	reaction to the fourth tour de franc won by miguel indurain .
c		indurain win tour reaction fourth tour franc won miguel indurain
d	-55.66	vincere tour reazione quarto tour francia
	-56.07	vincere tour reazione quarto giro francia
	-56.27	vincere tour reazione quarto tournee francia
	...	
e	-55.66	indurain vincere tour reazione quarto tour francia miguel indurain
	-56.07	indurain vincere tour reazione quarto giro francia miguel indurain
	-56.27	indurain vincere tour reazione quarto tournee francia miguel indurain
	...	
f		indurain vincere tour reazione quarto tour de france vincere miguel indurain
g		giro vittoria indurain reazione quarto giro de francia vincere miguel indurain

Table 3.7: Processing of English short topic 44. (a) Title and Descriptive fields of the original topic; (b) tokenized and stemmed query; (c) query after stop term removal; (d) *N*-best translations into Italian with log-probabilities; (e) translated queries after adding proper names; (f) corresponding human translated query; (g) corresponding translated query by Systran.

in Italian and in English, but names of locations and organizations often differ. An example is the name *Chechnya*, which in Italian is *Cecenia*. Finally, it is worth mentioning that proper names and numbers are excluded from the computation of the *N*-best translations, but are just added to them afterwards.

3.6.2 Blind Relevance Feedback

As queries are usually short, only few content words might be exploited for retrieval making the disambiguation among documents very hard. *Blind Relevance Feedback* (BRF) is a well known technique for adding other words related to the original queries that permits to improve retrieval performance. The basic idea is to perform retrieval in two steps. First, documents matching the original query **q** are ranked, then the R top ranked documents are taken and the T most relevant terms in them are added to the query. Hence, retrieval is repeated with the augmented query. In this work, new search terms are extracted from the R top documents according to [38]:

$$r_q \log \frac{(r_q + 0.5)(N - N_q - R + r_q + 0.5)}{(N_q - r_q + 0.5)(R - r_q + 0.5)} \tag{3.20}$$

where r_q is the number of documents, among the top R, which contain term q.

In CLIR, BRF is not used to expand the original query but its translations. In order to save computation time, BRF is performed on the N-best translation as a whole, as Equation (3.15) suggests. Hence, relevant terms of the top ranking documents are added to all N-best translations, without modifying their probabilities. Parameter setting of BRF was the same as for Italian monolingual IR.

3.7 Experiments

This Section reports performance of the presented IR systems on the CLEF tracks. In particular, monolingual IR experiments were carried out on Italian, while CLIR was performed from Italian to English and vice versa. Reported performance is in terms of mean average precision `mAvPr` (see Appendix B.1).

3.7.1 CLEF benchmark

CLEF tracks consists of collections of document and sets of topics, in different languages, and relevance assessments for every pair of document collection and set of topics (see Brachler and Peters, this volume). Three document collections of CLEF have been used here:

- an English collection (EC) consisting of 110,282 documents, from *Los Angeles Times* and issued in 1994;
- an Italian collection (IC1) including 58,051 documents, from *La Stampa* and issued in 1994 (used in CLEF 2000);
- an Italian collection (IC2) including IC1 and other 50,527 documents from the *Swiss News Agency*, issued in 1994, for a total of 108,578 documents (used in CLEF 2001 and CLEF 2002).

Topics consist in three fields (Title, Descriptive, Narrative), as shown in Table 3.6. Available topics are 40 for CLEF 2000, 50 for CLEF 2001, and 50 for CLEF 2002, for a total of 140. Table 3.8 reports statistics about document collections and topics. Notice that topics which do not have relevant documents in a given collection were removed from the corresponding track.
In the following, topics used for CLEF 2000, CLEF 2001, and CLEF 2002 will be referred to by Q1, Q2 and Q3, respectively. The language of each collection is indicated by the prefix of its name (I for Italian, E for English), while that of the queries depends on the considered track. Experiments were carried out using both short (TD) and long (TDN) topics.

3.7.2 Additional data

A commercial Italian-English dictionary of about 51K translation pairs is used, which the query-translation model relies on. On the average, each Italian term is translated by the dictionary into 1.84 English words, and vice-versa each

	Collection		
	EC	IC1	IC2
Documents	110,282	58,051	108,578
Size	425 MB	193 MB	278 MB

	Collection					
	EC		IC1		IC2	
	topics	rel.docs	topics	rel.docs	topics	rel.docs
Q1	33	579	34	338		
Q2	47	856			47	1246
Q3					49	1072

Table 3.8: Upper table: total numbers of documents and running words for each considered collection. Lower table: statistics about each pair of topic set and collection: i.e. number of topics with relevant documents in the collection, and total number of relevant documents in the collection.

English term has 1.68 Italian translations. The estimated coverage of the dictionary with respect to the query terms is 89.7% for Italian and 90.2% for English, including numbers which are translated verbatim.
The addition to the dictionary of the proper names found in the queries increased translation coverage to 94.6% for Italian words and 96.1% for English words.
BRF parameters R and T were estimated just for the Okapi model on some development data [7]. The best settings resulted $R = 5$ and $T = 15$.

3.7.3 Monolingual IR results

Table 3.9 reports performance achieved on each set of Italian topics and their union as well.
It can be noticed that performance on long topics is significantly better than on short ones. This is mainly due to the different number of content words which is available to search documents. Moreover, figures show that query expansion is very effective. Relative improvements due to BRF are between 8% and 22%

	Topics		Coll.	Statistical		Okapi		Combined	
Set	Type	Lang			+BRF		+BRF		+BRF
Q1	TD	IT	IC1	.3671	.4481	.4215	.4551	.4110	**.4556**
Q1	TDN	IT	IC1	.4447	.4941	.4920	**.5198**	.4722	.5152
Q2	TD	IT	IC2	.4141	.4662	.4449	.4815	.4379	**.4883**
Q2	TDN	IT	IC2	.4372	.4847	.4664	.4939	.4625	**.5041**
Q3	TD	IT	IC2	.3862	.4656	.4058	.4703	.4042	**.4920**
Q3	TDN	IT	IC2	.4453	.5271	.4432	.5028	.4516	**.5304**
Q1-2-3	TD	IT	IC1-2	.3913	.4612	.4240	.4704	.4182	**.4811**
Q1-2-3	TDN	IT	IC1-2	.4422	.5031	.4644	.5040	.4609	**.5169**

Table 3.9: Mean average precision results for monolingual IR, with different sets of Italian topics, topic types (TD vs. TDN), document collections (IC1, IC2, and both), and three retrieval models, each either with or without query expansion.

in the case of TD topics and between 5% and 10% for TDN topics. More precisely, performance improvements on the whole set of topics (Q1-2-3, both TD and TDN) result significant at level $p \geq 0.986$. It is worth noticing that BRF results more effective with the statistical LM approach than with the Okapi one.

A direct comparison between LM and Okapi shows that the latter performs slightly better, but differences become smaller after BRF. The last two columns of Table 3.9 report `mAvPr` results of the combined scoring model. Respectively, the columns correspond to the combination of scores taken before and after BRF on the single models. Figures show that after query expansion, the combined model, but in one case, improves over the best of the two single methods. Over the complete set of queries, relative improvements in mean-average precision over the best performing model are of 2.3% ($p \geq 0.984$) for short topics, and 2.6% ($p \geq 0.986$) for long topics.

	Topics		Coll.	1-best		5-best		10-best	
Set	Type	Lang			+BRF		+BRF		+BRF
Q1	TD	IT	EC	.3287	**.3463**	.3271	.3366	.3277	.3307
Q1	TDN	IT	EC	.3917	.4096	.3864	**.4391**	.3863	.4188
Q2	TD	IT	EC	.4593	.5035	.4537	**.5196**	.4532	.5128
Q2	TDN	IT	EC	.4934	.5132	.4977	**.5255**	.4737	.5226
Q1-2	TD	IT	EC	.4054	.4387	.4014	**.4441**	.4014	.4379
Q1-2	TDN	IT	EC	.4514	.4705	.4518	**.4899**	.4376	.4798

Table 3.10: Mean average precision results of Italian-English CLIR with the combined model. Experiments consider different sets of topics, topic types, always in Italian, one English target collection, and different numbers of *N*-best translations (1,5, and 10). Retrieval performance is reported either with or without blind relevance feedback.

3.7.4 Bilingual IR results

CLIR experiments were performed from Italian to English and in the opposite direction. In the CLEF evaluation campaigns, Italian-English tracks used topics Q1 and Q2, whereas English-Italian tracks used all sets of topics. Results of these runs are reported for each language direction in Tables 3.10 and 3.11, respectively. In all tracks, the combined method was used for the query-document model. Results are provided both for short and long topics, for each set of topics, and for their union. It is worth noticing that the Italian target collection changed between the first and the second CLEF campaign.

By looking at the results corresponding to different numbers of employed translations, it seems, at least on the average, that using more than one translation slightly improves performance. However, this conclusion is not confirmed from a statistical point of view. Only for the Italian-English task (after BRF, TDN topics), a significant difference in `mAvPr` between 5-best translations and 1-best translations was observed at level $p \geq .998$.

Considerations about query expansion, as stated for monolingual IR, are fully confirmed by the CLIR experiments.

	Topics		Coll.	1-best		5-best		10-best	
Set	Type	Lang			+BRF		+BRF		+BRF
Q1	TD	EN	IC1	.3125	**.3382**	.3192	.3339	.3068	.3180
Q1	TDN	EN	IC1	.3604	.3922	.3748	.4042	.3727	**.4135**
Q2	TD	EN	IC2	.3829	.4624	.3828	.4544	.3881	**.4691**
Q2	TDN	EN	IC2	.4156	.4851	.4184	.4849	.4235	**.4855**
Q3	TD	EN	IC2	.2993	.3444	.3086	.3531	.3161	**.3552**
Q3	TDN	EN	IC2	.3646	.4286	.3712	**.4410**	.3757	.4247
Q1-2-3	TD	EN	IC1-2	.3330	.3854	.3382	.3847	.3397	**.3866**
Q1-2-3	TDN	EN	IC1-2	.3819	.4395	.3892	**.4472**	.3922	.4438

Table 3.11: Mean average precision results of English-Italian CLIR with the combined model. Experiments consider different sets of topics, topic types, always in English, different Italian target collections (IC1, IC2, and both), and numbers of N-best translations (1,5, and 10). Retrieval performance is reported either with or without blind relevance feedback.

Further experiments were carried out to evaluate the query-translation model. In particular, CLIR experiments were performed by using query translations computed by the Viterbi search algorithm, by a commercial state-of-the-art machine translation system, and, finally, by a human. In the second case, the online Babelfish translation service, powered by Systran [32], was used. As Systran is supposed to work on fluent texts, preprocessing and translation steps were inverted in this case. As human translations, the topics in the documents' language were used, as provided by CLEF.

Given all topic translations, the CLIR algorithm for the 1-best case was applied. Results for Italian-English and English-Italian IR are reported in Tables 3.12 and 3.13, respectively.

Remarkably, the statistical query-translation method outperforms the Systran translation system on the union sets of topics. Significant differences between the two translation methods could only be measured on the English-Italian retrieval task. Differences were significant at level $p \geq 0.96$ on short topics, and at level $p \geq 0.76$ on long topics.

Topics			Coll.	Translation		
Set	Type	Lang		Systran	1-best	human
Q1	TD	IT	EC	**.4007**	.3463	.4866
Q1	TDN	IT	EC	**.4565**	.4096	.5029
Q2	TD	IT	EC	.3900	**.5035**	.5559
Q2	TDN	IT	EC	.4786	**.5132**	.5703
Q1-2	TD	IT	EC	.3944	**.4387**	.5273
Q1-2	TDN	IT	EC	.4695	**.4705**	.5425

Table 3.12: Mean average precision results of Italian-English CLIR with the combined model including query expansion. Experiments consider different sets of Italian topics, topic types, one English collection, and different kinds of translations: computed by Systran, the 1-best statistical model, and human made.

From both Tables 3.12 and 3.13 it is evident that IR results with 1-best translations shows more oscillations around the global `mAvPr` value computed over the union sets of topics. To investigate this issue, standard deviations of the average precision were computed over the whole set of topics, for each experimental condition. On the Italian-English track, standard deviations with TD topics were .314 and .298, respectively, for 1-best and Systran translations. On TDN topics, standard deviations were exactly the same, .301 for both translation methods. On the English-Italian track, 1-best translations seem to cause even less variability than the Systran ones: on TD topics, standard deviations of .323 and .331 were respectively measured, while on TDN topics the corresponding standard deviations were .316 and .329.

Unfortunately, these measurements confirm the difficulty of finding some statistically meaningful explanation of the different `mAvPr` behavior of the tested systems over the single sets of topics.

A problem in translating topics is that some random noise is introduced in the retrieval process. Erroneous translations of content words may indeed severely affect retrieval performance and, in general, the loss in performance is not strictly

Topics			Coll.	Translation		
Set	Type	Lang		Systran	1-best	human
Q1	TD	EN	IC1	.3378	**.3382**	.4556
Q1	TDN	EN	IC1	.3781	**.3922**	.5152
Q2	TD	EN	IC2	.3637	**.4624**	.4883
Q2	TDN	EN	IC2	.3872	**.4851**	.5041
Q3	TD	EN	IC2	**.4037**	.3444	.4920
Q3	TDN	EN	IC2	**.4412**	.4286	.5304
Q1-2-3	TD	EN	IC1-2	.3720	**.3854**	.4811
Q1-2-3	TDN	EN	IC1-2	.4052	**.4395**	.5196

Table 3.13: Mean average precision results of English-Italian CLIR with the combined model including query expansion. Experiments consider different sets of English topics, topic types, different document collections, and different kinds of translations: computed by Systran, the 1-best statistical model, and human made.

related to the number of translation errors.

An indication about the noise introduced by the translation process comes from the lower standard deviations which can be measures on the retrieval results with human translations: .287 for TD topics and .279 for TDN topics, in the Italian-English track, and .300 for TD topics and .289 on TDN topics, in the English-Italian track. Hence, in general, automatic translation increases uncertainty in `mAvPr`, which can be quantified in 4%-10% relative increase of standard deviation.

In our statistical model, the chance of correctly translating a content word, in a given context, depends on several nested events: the dictionary contains the word, the right translation is among the ones available for that word, and, finally, the correct one is selected. In the following, a qualitative analysis of translation errors is presented.

3.7.5 Qualitative evaluation

A qualitative analysis of results was carried out to better understand possible weak points of the statistical query translation method. Differences in average precision (AvPr) achieved on each single topic were computed. The resulting plots are shown for both translation directions in Figures 3.2 and 3.3, respectively. More specifically, results refer to the combined model, using short topics and no BRF.

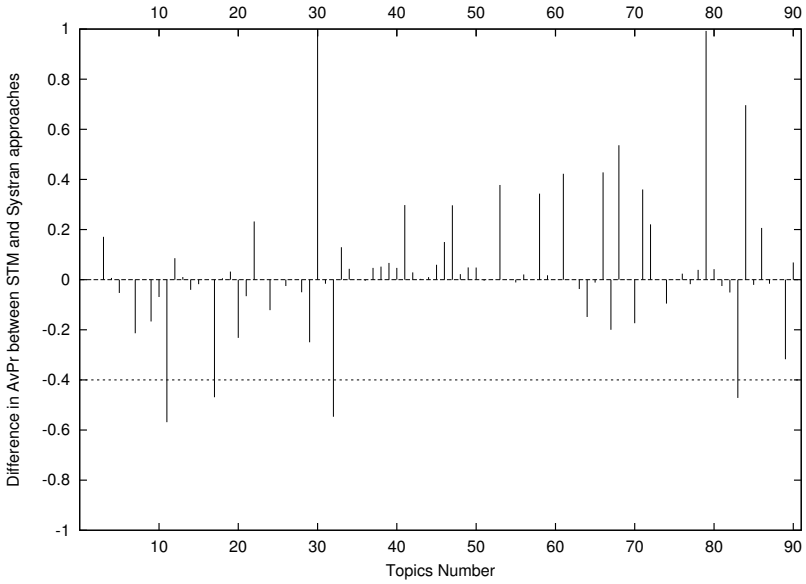

Figure 3.2: Differences in average precision corresponding to the same CLIR system using translations computed either by the statistical model or by a commercial machine translation system. Topics are in Italian, documents in English.

It results that the two translation approaches achieve similar performance for most of the topics, in fact, only 10% of them show AvPr differences higher than 0.4. Hence, a more detailed analysis was made on the subset of topics on which Systran translations performed significantly better. Topics 11, 17, 32, and 83 for Italian-English CLIR and topics 15, 35, and 126 for English-Italian CLIR were considered.

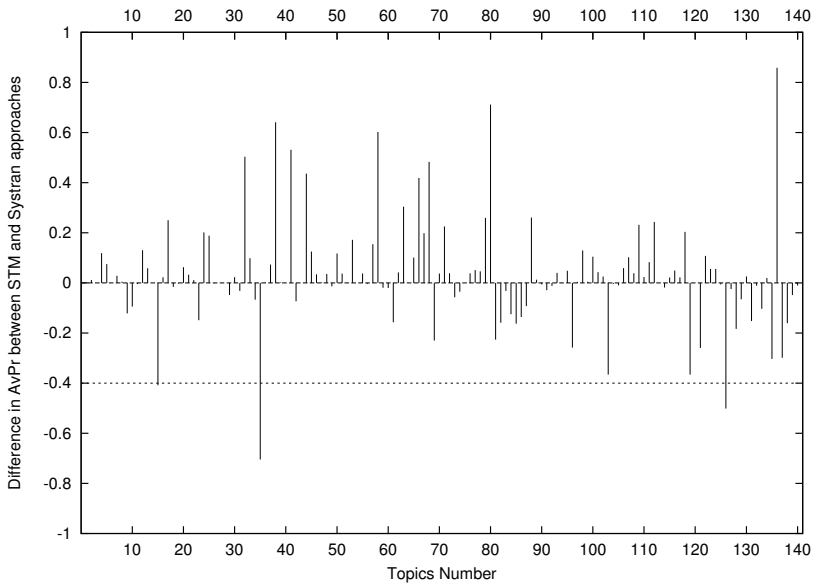

Figure 3.3: Differences in average precision corresponding to the same CLIR system using translations computed either by the statistical model or by a commercial machine translation system. Topics are in English, documents in Italian.

Poor performance generally resulted from translations errors of content words. Translation errors were either caused by a wrong analysis at the POS tagging or word stemming levels, or by coverage failures of the bilingual dictionaries. For instance, the Italian word *preti* (*priests*) was not correctly transformed into its singular form *prete* (*priest*); similarly, the English word *wolves* was wrongly stemmed as *wolv*, instead of *wolf*. Hence, in both cases the corresponding entries were not found in the bilingual dictionary. Bad retrieval also occurred because correct translations appeared with a low rank; e.g. the best Italian translation of *fur* (*pelliccia*) appears for the first time in the translation at rank 22.

Chapter 4

Text Translation

In this Chapter, we present novel statistical models which extend the well known machine translation (MT) framework developed by [12]. Accordingly, the most likely translation of a source sentence $\mathbf{f}$ can be searched for by ranking strings $\mathbf{e}$ in the target language[1] according to the product of the language model probability $\Pr(\mathbf{e})$ and the translation model probability $\Pr(\mathbf{f} \mid \mathbf{e})$. While the first factor is usually computed through an n-gram language model [36], six translation (or word-alignment) models of increasing complexity have been proposed in [12, 60], which are conventionally numbered from 1 to 6. The complexity of the search procedure basically depends on the way words of the source string can be translated and re-ordered in the target string. Given that even the simplest word-alignment model causes an exact search to be NP-complete [40], approximation algorithms [18] have been proposed, in particular for the so called Model 4, which is the model considered in this Chapter. For instance, in [79] a polynomial search algorithm for Model 4 has been derived from a *dynamic programming* (DP) solution of the traveling-salesman problem. Instead, we propose a different DP-based algorithm which directly derives from the search criterion.

Besides the success of word-alignment models, several attempts have been pur-

[1] We follow here the notation of [12], which assumes as source and target languages, French and English, respectively .

sued to overcome some of their shortcomings, especially the little use of context within the source string. Recently, several research labs have reported improvements in translation accuracy by shifting from word- to phrase-based MT [47, 90, 91, 84, 41, 78]. Best performing approaches rely on a sample of *phrase*[2] pairs, which is automatically extracted from a word-aligned parallel corpus. Moreover, ad-hoc statistical models and search algorithms are developed, which explicitly consider possible segmentations of the source string at phrase level. Instead, we propose phrase-based translation models which are tightly related to Model 4. Basically, after augmenting the target language vocabulary with a list of phrases, model parameters are estimated in one of three possible ways: (i) from an available word-based model, (ii) through statistics extracted from a sample of phrase pairs, (iii) by combining the two previous methods. By defining the target language over the augmented vocabulary, the same search algorithm for Model 4 can be deployed.

The Chapter is organized as follows. Section 4.1 overviews and acknowledges previous results our work follows along or departs from. Section 4.2 shortly reviews basic notations and concepts related to the statistical MT framework, in general, and to the here used word-based model, in particular. Section 4.3 introduces the search problem in a formal way and derives the dynamic programming based MT algorithm implemented in our system. Section 4.4 discusses the extraction of phrase-pairs from a parallel corpus, and presents extensions of the statistical MT approach to account for translation into target phrase sequences. Section 4.6 reports comparative experiments between different word- and phrase-based translation settings. Finally, Section 7.2 reports conclusions and indicates directions of future work.

[2]Following a widespread as well as disputable habit, we mean here by *phrase* any finite word sequence, either in the source or target language, regardless of its linguistic soundness.

4.1 Previous work

The first search algorithm based on statistical word-alignment models appeared in a US patent authored by [11]. In particular, an A* (or stack) decoding algorithm for Model 3 was proposed which incrementally extends partial translations of the source string, until an optimal full translation is found. To reduce the search complexity, the number of active hypotheses is limited as well as the possible word reordering. In particular, active hypotheses of different length are stored into separate stacks, so that pruning criteria are applied on comparable hypotheses, i.e. of the same length. Improvements on the stack decoding approach were proposed for Model 2 by [88], through the introduction of linguistically motivated constraints on word reordering. On the other hand, [85] derived a more efficient stack decoding algorithm by using a simplified translation model.

In [29] a *greedy* decoding algorithm for Model 4 was presented which incrementally tries to improve an initial translation guess of the source sentence. At each step, new solutions are checked by applying a number of local modifications to the current solution.

The first search algorithm based on dynamic programming is reported by [80], which considered an augmented version of Model 2. Subsequently, [79] developed a search algorithm for Model 4, by combining a dynamic programming solution of the traveling-salesman problem with reordering constraints defined by a finite state automaton. The dynamic programming approach results similar to the A* algorithm with the crucial difference that *recombination* is applied on hypotheses sharing similar substructure. In addition, a beam search technique is applied to limit the set of active hypotheses. The here presented search algorithm goes along the way paved by [79], but relies on a different and simpler, to our view, decomposition of the optimization criterion.

More recently, there has been a tendency to augment stochastic dependencies

beyond the limit imposed by the word-based models. In [61], the so-called *alignment template* approach is introduced which directly models translation at the phrase level. The search algorithm, which eventually relies on a preliminar phrase-segmentation of the input, exploits a collection of translation phrases, with alignment information, extracted from the corpus.

Several methods to extract translation phrases (or templates) from parallel corpora have been proposed, either based on alignment patterns [61, 47, 41, 84, 78] or on linguistic structures [90].

Following [61], ad-hoc phrase-bases statistical translation models and decoding algorithms were developed, that exploit statistics at phrase and word levels. Most of these models can be interpreted as attempts to unify memory- and statistical-based approaches. For instance, in [47], the greedy algorithm in [29] is modified by exploiting an additional translation guess computed by recognizing and translating phrases of the source string. In [91], a translation model just based on phrase translation frequencies is discussed and embedded into a search algorithm that takes into account possible phrase-segmentations and local re-orderings. In [41], a similar model is discussed which also embeds a simple distorsion model to account for phrase reordering. Finally, in [48, 78] joint probability phrase models are introduced, which assume the target and source strings are generated progressively and simultaneously.

In contrast to all the above mentioned approaches, our phrase-based translation models and search algorithm are tightly related and basically extend word Model 4. In this way, we believe, our models embed the advantages of using a wider context in translation and, at the same time, preserve the robustness of the word-based approach in conditions of data sparseness.

4.2 Statistical Machine Translation

From a statistical point of view MT can be seen as the problem of finding the sentence in the target language which maximizes a given probability distribution $\Pr(\mathbf{e} \mid \mathbf{f})$[3]. This probability measure should measure how good the target sentence translates the source string. Formally, the Statistical Machine Translation (SMT) problem can be stated as follows: given a string $\mathbf{f}$ in a source language $\mathcal{F}^*$, find the string $\mathbf{e}^*$ in the target language $\mathcal{E}^*$, which maximizes the probability distribution $\Pr(\mathbf{e} \mid \mathbf{f})$:

$$\mathbf{e}^* = \arg\max_{\mathbf{e} \in \mathcal{E}^*} \Pr(\mathbf{e} \mid \mathbf{f}) \tag{4.1}$$

The definition of the probability measure $\Pr(\mathbf{e} \mid \mathbf{f})$ is the main issue related to SMT. The measure should be able to model at the best all aspects of the translation process, adequacy and fluency above all.

An SMT model usually features a lot of parameters, and an efficient training procedure for their optimization is strongly recommended.

In the statistical framework any string in the target language is a possible translation of the input sentence. Even limiting the length of the output strings to twice the length of the input, the number of hypotheses is very huge. Hence, the search algorithm for finding out the best one should be very efficient; to do that it should exploit at the best the features of the model.

Historically first attempts to tackle out SMT were based on the so-called *source-channel approach* introduced in the AI field by Shannon [73]. This framework assumes that the string $\mathbf{f}$ is obtained as a result of the modifications which the original string $\mathbf{e}$ has undergone. As shown in Figure 4.1 the process of translation, a string $\mathbf{e}$ is first generated by a source described by the model $\Pr(\mathbf{e})$ and then modified into $\mathbf{f}$ through a channel described by a model $\Pr(\mathbf{f} \mid \mathbf{e})$. Finally,

[3]The notational convention will be as follow throughout all the thesis. The symbol $\Pr(\cdot)$ is used to denote general probability distributions with (nearly) no specific assumptions. In contrast, for model-based probability distributions, the generic symbol $p(\cdot)$ is used.

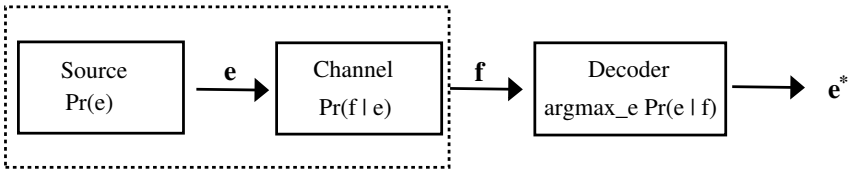

Figure 4.1: Translation process in the source-channel approach.

the decoder has to find the best output $\mathbf{e}^*$ which best matches to the original $\mathbf{e}$. Formally, this approach is based on the exact Bayes decomposition:

$$\Pr(\mathbf{e} \mid \mathbf{f}) = \frac{\Pr(\mathbf{e}) \Pr(\mathbf{f} \mid \mathbf{e})}{\Pr(\mathbf{f})} \tag{4.2}$$

causing the modification of the SMT problem into the following search criterion:

$$\mathbf{e}^* = \arg\max_{\mathbf{e}} \frac{\Pr(\mathbf{e}) \Pr(\mathbf{f} \mid \mathbf{e})}{\Pr(\mathbf{f})} \tag{4.3}$$

$$= \arg\max_{\mathbf{e}} \Pr(\mathbf{e}) \Pr(\mathbf{f} \mid \mathbf{e}) \tag{4.4}$$

where the denominator $\Pr(\mathbf{f})$ is discarded because of its constancy with respect to the maximization variable.

The source-channel approach factors out the SMT model $\Pr(\mathbf{e} \mid \mathbf{f})$ into two models, briefly described in the rest of this Section: the *language model* $\Pr(\mathbf{e})$ is devoted to model the target string features like the fluency; *the translation model* $\Pr(\mathbf{f} \mid \mathbf{e})$ is mostly devoted to describe the transfer of the meaning from the input to the output sentence.

The criterion (4.3) can be shown to be optimal if the true probability distributions are used. But only poor approximations of them can reasonably obtained, and a different combination of the language and translation models might yield better results. Hence, the criterion should modify with the introduction of two exponential weights λ_{lm} and λ_{tm} as follows:

$$\mathbf{e}^* = \arg\max_{\mathbf{e}} \Pr(\mathbf{e})^{\lambda_{lm}} \Pr(\mathbf{f} \mid \mathbf{e})^{\lambda_{tm}} \tag{4.5}$$

A second approach, becoming famous at the end of '90s, defines directly $\Pr(\mathbf{e} \mid \mathbf{f})$ by means of a log-linear model. This framework will be presented in the next Chapter.

4.2.1 Language model

Language modeling in statistical MT is tightly related to the way the search algorithm works. Besides computing the probability of complete translation hypotheses $\mathbf{e}$, it is desirable to evaluate language model scores even on partial translation hypotheses, so that they can be properly ranked. For this purpose, in the here discussed search algorithms, n-gram language models [36] are deployed. In particular, trigram probabilities are estimated by interpolating non-linearly smoothed trigram, bigram, and unigram frequencies [25]. Moreover, n-gram frequency statistics are collected over a large target language corpus, including the target language sentences used to estimate the translation model. Given a target string $\mathbf{e} = e_1^l$, the trigram language model probability is:

$$\Pr(\mathbf{e}) = p(l)\ p(\mathbf{e} = e_1^l \mid l) = p(l) \prod_{i=1}^{l} p(e_i \mid e_{i-2}, e_{i-1}) \tag{4.6}$$

where $p(l)$ is a probability distribution of the string length l to ensure that Equation (4.6) defines a probability measure over the set of finite strings of the target vocabulary $\mathcal{E}$. In particular, we assume a uniform length distribution over a support set of target strings of length at most L, where L is twice the maximum allowed length for the input string $\mathbf{f}$. In order to let the LM better predict initial and final words of a sentence, special boundary symbols are put as first and last words of the source sentence. These symbols are enforced to be translated verbatim in the target sentence.

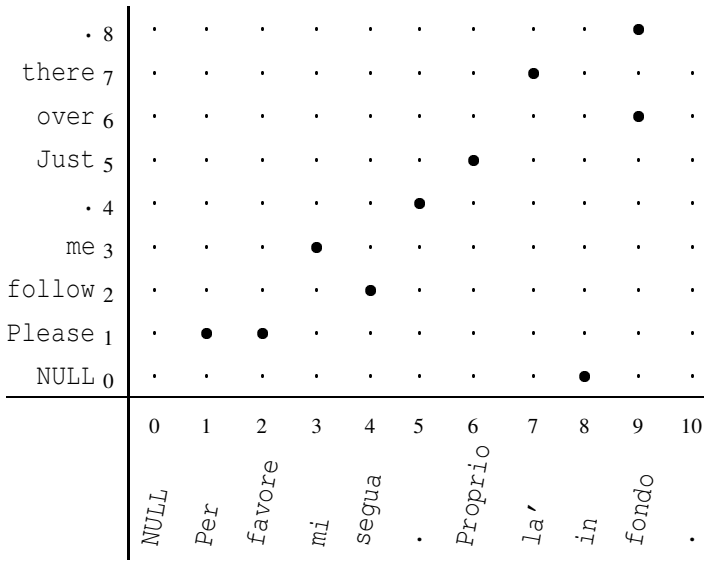

Figure 4.2: Example of translation pair and alignment, with source in Italian and target in English. Alignments link each source position either to one target position or to the empty word (NULL) at position zero. Alignments do not necessarily cover all target positions.

4.2.2 Alignments

Many SMT models rely on the concept of *alignment*, which is first introduced by [12]. Given a pair of sentence $\mathbf{f} = f_1^m = f_1, \ldots, f_m$ in French and $\mathbf{e} = e_1^l = e_1, \ldots, e_l$ in English, an alignment between them is "an object indicating for each word in the French string that word in the English string from which it arose". It is useful to add an English empty word $e_0 = \texttt{NULL}$, which represents the virtual ending point for those words in $\mathbf{f}$ not connected to any English words. In the same way $f_0 = \texttt{NULL}$ is defined. As connection between f_0 and e_0 is absolutely meaningless, it will never be set.

In a mathematically sound framework, we define an alignment $\mathbf{a}$ as a relation between words in $\mathbf{f}$ and $\mathbf{e}$: $\mathbf{a} = \mathbf{f}\mathcal{R}\,\mathbf{e} \subseteq \{(f_j, e_i) \mid j = 0, \ldots, m,\ i = 0, \ldots, l\}$. Graphically an alignment can be represented by means of a matrix, where the entry (i, j) is bulleted if there is a *connection* between f_j and e_i, i.e. if f_j is aligned with e_i, as shown in Figure 4.2.

In general each French word can be connected to any English word, and vicev-

ersa; henceforth, the number of possible alignments between $\mathbf{f}$ and $\mathbf{e}$ is $2^{(m+1)(l+1)}$.

Two classes of alignments are usually taken into a account. An alignment between **f** and **e** is called *direct* if there exists exactly one connection exiting from real French words, and no connections exit f_0; it results that a direct alignment is a function from $\mathbf{f} = f_0^m = f_0, \ldots, f_m$ to $\mathbf{e} = e_0^l = e_0, \ldots, e_l$ with the constraint that f_0 is virtually aligned only to e_0. Viceversa an alignment between **f** and **e** is called *inverted*, if the corresponding alignment between **e** and **f** is direct; in this case the alignment is a function from **e** to **f** with the constraint that e_0 is virtually aligned only to f_0. The alignment represented in Figure 4.2 is direct.
The number of possible direct and inverted alignments (m^{l+1} and l^{m+1}, respectively) is significantly lower, although even huge.
Smaller subset of alignments can be defined by imposing constraints over the legal connections between **f** and **e**. An alignment satisfying a given set of constraints is called *compatible* with those constraints. In the following we will assume to have specific constraints, and to consider only the set $\mathcal{A}(\mathbf{f}, \mathbf{e})$ of the compatible alignments.

4.2.3 Translation models

According to [12], given a French (source) string **f** and an English (target) string **e**, the translation probability $\Pr(\mathbf{f} \mid \mathbf{e})$ is expressed by the marginal probability:

$$\Pr(\mathbf{f} \mid \mathbf{e}) = \sum_{\mathbf{a} \in \mathcal{A}(\mathbf{f}, \mathbf{e})} \Pr(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) \tag{4.7}$$

where the *hidden* variable **a** consists of an alignment from source to target positions, and $\Pr(\mathbf{f}, \mathbf{a} \mid \mathbf{e})$ is a suitable translation model. Besides the definition of the probability measure a translation model imposes constraints to limit the size of the set $\mathcal{A}(\mathbf{f}, \mathbf{e})$.
In [12, 60], six translation models of increasing complexity are introduced, referred to as Model 1 to Model 6. In the next subsection we briefly review Model

4, because the phrase-based translation mdoel we define is partially related to it.

4.2.4 Model 4

Given the string $\mathbf{e} = e_1^l = e_1, \ldots, e_l$, a string $\mathbf{f}$ and an alignment $\mathbf{a}$ are generated as follows:

i. a non-negative integer ϕ_i, called *fertility*, is generated for each word e_i and for the null word e_0;

ii. for each e_i, a list τ_i, called *tablet*, of ϕ_i source words and a list π_i, called *permutation*, of ϕ_i source positions are generated;

iii. finally, if the generated permutations define a compatible alignment from $\mathbf{f}$ to $\mathbf{e}$ then the process succeeds, otherwise it fails.

Fertilities fix the number of source words to be aligned to each target word, and the total length m of the French string: $m = \sum_{i=0}^{l} \phi_i$. Target words (including e_0) with positive fertility are called *cepts*.
If a phrase $\tilde{e}_i$ has fertility zero, i.e. $\phi_i = 0$, it means that $\tilde{e}_i$ is not associated with any input word, but is simply introduced to get a more fluent translation.
Model 4 imposes that the alignment between $\mathbf{f}$ and $\mathbf{e}$ is direct; it results that sets π_i are mutually disjuncted, i.e. $\pi_i \cap \pi_j = \emptyset$ for all $i \neq j$, and define a partition of the string $\mathbf{f}$. Moreover, permutations of Model 4 are constrained to assign positions in ascending order. Details of other finer contraints can be found in [12].
Taking into account these constraints, it can be shown that if the process succeeds in generating a triple $(\phi = \phi_0^l, \tau = \tau_0^l, \pi = \pi_0^l)$[4] then there is exactly one corresponding pair $(\mathbf{f}, \mathbf{a})$, and vice versa. This property justifies the following

[4] In the following, we will assume that arrays array $\mathbf{e}, \mathbf{a}, \phi, \tau, \pi$ have length $l+1$, if not differently specified.

definition of Model 4:

$$\Pr(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) = p(\phi, \tau, \pi \mid \mathbf{e}, l) \tag{4.8}$$

$$= p(\phi_0 \mid m') \prod_{i=1}^{l} p(\phi_i \mid e_i) \prod_{i=0}^{l} p(\tau_i \mid \phi_i, e_i) \frac{1}{\phi_0!} \prod_{i=1}^{l} p(\pi_i \mid \phi_i, \bar{\pi}_i) \tag{4.9}$$

$$= p(\phi_0, \tau_0 \mid m') \prod_{i=1}^{l} p(\phi_i, \tau_i, \pi_i \mid e_i, \bar{\pi}_i) \tag{4.10}$$

where l is the length of $\mathbf{e}$, $m' = m - \phi_0 = \sum_{i=1}^{l} \phi_i$, and $\bar{\pi}_i$ is the *center* [5] of the most recent cept. It is worth remarking that $\pi_0 = -1$ by default and $\pi_i = \pi_{i-1}$ if $\phi_i = 0$.

In the right side of Equation (4.9) we can identify three models, namely *fertility*, *lexicon*, and *distortion* models.

Fertility model represents step (i) of the generative process; fertilities ϕ_0^l of words e_1^l are generated through $p(\phi \mid e)$, while the fertility ϕ_0 of e_0 is generated through a Binomial distribution $p(\phi \mid m')$.

$$p(\phi \mid \mathbf{e}, l) = \prod_{i=0}^{l} p(\phi_i \mid e_i) = p(\phi_0 \mid \sum_{i=1}^{l} \phi_i) \prod_{i=1}^{l} p(\phi_i \mid e_i) \tag{4.11}$$

$$p(\phi_i \mid e_i) = \begin{cases} p(\phi_0 \mid m') \\ p(\phi_i \mid e_i) \quad 1 \leq i \leq l \end{cases} \tag{4.12}$$

During step (ii) the lexicon model (4.13) generates tablets for cepts through the following distribution:

$$p(\tau \mid \phi, \mathbf{e}, l) = \prod_{i=0}^{l} p(\tau_i \mid \phi_i, e_i) \tag{4.13}$$

$$p(\tau_i \mid \phi_i, e_i) = \prod_{k=1}^{\phi_i} p(\tau_{i,k} \mid e_i) \quad 0 \leq i \leq l \tag{4.14}$$

The distortion model generates permutations π_1^l relying on two probability tables:

[5] The center of a cept is defined as the ceiling of the mean position assigned to it.

- $p_{=1}(\Delta)$, which considers the distance Δ between the first generated position and the *center*[6] of the most recent cept;
- $p_{>1}(\Delta)$, which considers the distance Δ between any two consecutively assigned positions of the permutation.

Finally, positions for e_0 are generated at random over the residual ϕ_0 positions, with probability $\frac{1}{\phi_0!}$.

$$p(\pi \mid \phi, l) = \prod_{i=0}^{l} p(\pi_i \mid \phi_i, \bar{\pi}_i) = \frac{1}{\phi_0!} \prod_{i=1}^{l} p(\pi_i \mid \phi_i, \bar{\pi}_i) \tag{4.15}$$

$$p(\pi_i \mid \phi_i, \bar{\pi}_i) = \begin{cases} \dfrac{1}{\phi_0!} & \\ p_{=1}(\pi_{i,1} - \bar{\pi}_i) \displaystyle\prod_{k=2}^{\phi_i} p_{>1}(\pi_{i,k} - \pi_{i,k-1}) & 1 \leq i \leq l \end{cases} \tag{4.16}$$

The here considered distortion model omits some dependencies specified in [12].

Moreover, it is worth remarking that English words with fertility zero may only generate an empty tablet and an empty permutation with probability 1, i.e.:

$$p(\tau_i \mid \phi_i = 0, e_i) = \begin{cases} 1 & \text{if } \tau_i = \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{4.17}$$

$$p(\pi_i \mid \phi_i = 0, \bar{\pi}_i) = \begin{cases} 1 & \text{if } \pi_i = \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{4.18}$$

In order to simplify notation we introduce:

$$p(\phi_0, \tau_0 \mid \sum_{i=1}^{l} \phi_i) = p(\phi_0 \mid \sum_{i=1}^{l} \phi_i)\ p(\tau_0 \mid \phi_0, e_0)\ \frac{1}{\phi_0!} \tag{4.19}$$

$$p(\phi_i, \tau_i, \pi_i \mid e_i, \bar{\pi}_i) = p(\phi_i \mid e_i)\ p(\tau_i \mid \phi_i, e_i)\ p(\pi_i \mid \phi_i, \bar{\pi}_i) \quad 1 \leq i \leq l \tag{4.20}$$

[6] $\bar{\pi}_i$ is defined as the ceiling of the mean position assigned to the most recent cept before i.

which has to be interpreted as the probability of choosing fertility ϕ, tablet τ and permutation π for the empty word e_0 and e_i, respectively. Moreover, we define

$$p(\phi, \tau, \pi \mid \mathbf{e}, i) \;=\; p(\phi_0, \tau_0 \mid m - \phi_0) \prod_{s=1}^{i} p(\phi_i, \tau_i, \pi_i \mid e_i, \bar{\pi}_i) \qquad (4.21)$$

representing the probability of a partial solution of length i.
Notice that if we consider the generative process described in Section 4.2.4, based on the source-channel approach, the previous quantities are not well defined because $m = \sum_{i=0}^{l} \phi_i$ only after all fertilities are chosen. But from the point of view of the translation we know the value m at the beginning, because it is the length of input. Hence, $p(\phi, \tau, \pi \mid \mathbf{e}, i)$ can be used during the search

4.2.5 Training

Parameters γ of the language model $p_\gamma(\mathbf{e})$ and θ of the translation model $p_\theta(\mathbf{f} \mid \mathbf{e})$ just introduced are trained through the Maximum Likelihood (ML) criterion. If $\{(\mathbf{f}_s, \mathbf{e}_s) : s = 1, \ldots, S\}$ is a parallel corpus of S aligned senteces and $\{\mathbf{e}_t : t = 1, \ldots, T\}$ is a monolingual corpus of T sentences, this criterion is expressed as follows:

$$\hat{\gamma} \;=\; \arg\max_{\gamma} \prod_{t=1}^{T} p_\gamma(\mathbf{e}_t) \qquad (4.22)$$

$$\hat{\theta} \;=\; \arg\max_{\theta} \prod_{s=1}^{S} p_\theta(\mathbf{f}_s \mid \mathbf{e}_s) \qquad (4.23)$$

While optimal language model parameters $\hat{\gamma}$ are estimated with a LM software developed at ITC-irst, the maximization of translation model parameters is performed through the open source toolkit GIZA++ [60]. This software exploits a suitable parallel corpus of sentence pairs through the Expectation Maximization (EM) algorithm [21] or some approximations of it.
Use of exponential weights introduced in Section 4.2 can be generalized to consider the three sub-models of the translation model $p_\theta(\mathbf{f} \mid \mathbf{e})$. Thus, instead of

only one weight λ_{tm}, λ_{fert}, λ_{lex} and λ_{dist} can be used for weighting the fertility model $p(\phi \mid \mathbf{e}, l)$, the lexicon model $p(\tau \mid \phi, \mathbf{e}, l)$, and distortion model $p(\pi \mid \phi, l)$, respectively.
Optimization of these weights, which might improve system performance, can be achieved through a procedure described in next Chapter.
For simplifying notation, weights are not used in the rest of the Chapter.

4.3 Search problem

By exploiting (4.7) for the translation model $\Pr(\mathbf{f} \mid \mathbf{e})$, the search criterion (4.3) becomes:

$$\mathbf{e}^* = \arg\max_{\mathbf{e}} \Pr(\mathbf{e}) \Pr(\mathbf{f} \mid \mathbf{e}) \tag{4.24}$$

$$= \arg\max_{\mathbf{e}} \sum_{\mathbf{a} \in \mathcal{A}(\mathbf{f}, \mathbf{e})} \Pr(\mathbf{e}) \Pr(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) \tag{4.25}$$

As the amount of alignments in $\mathcal{A}(\mathbf{f}, \mathbf{e})$ can be very huge, the following approximate criterion is defined:

$$\mathbf{e}^* \approx \arg\max_{\mathbf{e}} \max_{\mathbf{a} \in \mathcal{A}(\mathbf{f}, \mathbf{e})} \Pr(\mathbf{e}) \Pr(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) \tag{4.26}$$

The substitution of the summation with the maximum operation permits the implementation of an efficient algorithm based on Dynamic Programming (DP) paradigm.
Now we derive the formulation of a DP-based search algorithm which exploits model 4 and the previous approximate criterion.
Given the source sentence $\mathbf{f} = f_1^m$, a corresponding translation $\mathbf{e}^*$ is searched as follows:

$$\mathbf{e}^* \approx \arg\max_{\mathbf{e}} \Pr(\mathbf{e}) \max_{\mathbf{a} \in \mathcal{A}(\mathbf{f}, \mathbf{e})} \Pr(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) \tag{4.27}$$

$$= \arg\max_{l, \mathbf{e}} p(\mathbf{e} \mid l) \max_{(\phi, \tau, \pi) \in \mathcal{A}(\mathbf{f}, \mathbf{e})} p(\tau, \pi, \phi \mid \mathbf{e}, l) \tag{4.28}$$

$$= \arg\max_{l, \mathbf{e}} p(\mathbf{e} \mid l) \max_{\pi \in \mathcal{P}(\{1, \ldots, m\})} p(\tau, \pi, \phi \mid \mathbf{e}, l) \tag{4.29}$$

where the constraint $(\phi, \tau, \pi) \in \mathcal{A}(\mathbf{f}, \mathbf{e})$ means that fertilities, tablets and permutations must be compatible with word alignments between $\mathbf{f}$ and $\mathbf{e}$, i.e. the corresponding source-alignment pair $(\mathbf{f}, \mathbf{a})$ has to cover all the m source positions exactly once. In Equation (4.29), a shorter but equivalent notation is used for this constraint: as τ and ϕ are univocally determined from π and $\mathbf{f}$, we just constrain π to define a partition over the set of source positions, i.e. $\pi \in \mathcal{P}(\{1, \ldots, m\})$.

The score Q^* of the optimal translation $\mathbf{e}^*$ can be computed by explicitly searching among optimal solutions of fixed length, i.e.:

$$Q^* = \max_{l,\mathbf{e}} p(\mathbf{e} \mid l) \max_{\pi \in \mathcal{P}(\{1,\ldots,m\})} p(\phi, \tau, \pi \mid \mathbf{e}, l) = \max_l Q_l^* \tag{4.30}$$

Further, the optimal score of a solution of length l can be searched among solutions which also fix the last two words and the center of the last cept, i.e.:

$$\begin{aligned} Q_l^* &= \max_{e',e} \max_{\bar{\pi}} \max_{\mathbf{e}: e_{l-1}=e', e_l=e} p(\mathbf{e} \mid l) \max_{\pi \in \mathcal{P}(\{1,\ldots,m\}): \bar{\pi}_l = \bar{\pi}} p(\phi, \tau, \pi \mid \mathbf{e}, l) \\ &= \max_{e',e} \max_{\bar{\pi}} Q_l(\{1, \ldots, m\}, \bar{\pi}, e', e) \end{aligned} \tag{4.31}$$

The above criterion permits us to introduces the quantity $Q_i(\mathcal{C}, \bar{\pi}, e', e)$ which indicates the score of an optimal solution of length i, with last target words e', e, center of the last cept $\bar{\pi}$, and permutations π_0^i defining a partition of the subset of source positions $\mathcal{C}$. This quantity can be defined recursively with respect to the length i as follows:

base: $i = 0$

$$Q_0(\pi_0, \bar{\pi}_0, \varepsilon, \varepsilon) = p(\phi, \tau, \pi \mid \mathbf{e}, 0) = p(\phi_0, \tau_0 \mid m - \phi_0) \tag{4.32}$$

step: $i > 0$

$$\begin{aligned} Q_i(\mathcal{C}, \bar{\pi}, e', e) &= \max_{e_1^i: e_{i-1}=e', e_i=e} p(\mathbf{e} \mid i) \max_{\pi_0^i \in \mathcal{P}(\mathcal{C}): \bar{\pi}_i = \bar{\pi}} p(\phi, \tau, \pi \mid \mathbf{e}, i) \\ &= \max_{e''} p(e \mid e'', e') \times \end{aligned} \tag{4.33}$$

$$\max \begin{cases} \max\limits_{\emptyset \subset \pi_i \subseteq C : \bar{\pi}_i = \bar{\pi}} \max\limits_{\bar{\pi}'} p(\phi_i, \tau_i, \pi_i \mid e, \bar{\pi}') \times \\ \qquad Q_{i-1}(C \setminus \pi_i, \bar{\pi}', e'', e') \\ p(\phi_i = 0 \mid e)\ Q_{i-1}(C, \bar{\pi}, e'', e') \end{cases} \tag{4.34}$$

$$= \max_{e''} \max_{\emptyset \subseteq \pi_i \subseteq C : \bar{\pi}_i = \bar{\pi}} \max_{\bar{\pi}'} p(e \mid e'', e')\ p(\phi_i, \tau_i, \pi_i \mid e, \bar{\pi}') \times Q_{i-1}(C \setminus \pi_i, \bar{\pi}', e'', e') \tag{4.35}$$

Last step is trivially correct by remembering that $p(\phi_i, \tau_i, \pi_i \mid e, \bar{\pi}') = p(\phi_i = 0 \mid e)$ if $\pi_i = \emptyset$, as derived from Equations (4.11, 4.13, 4.15, 4.20).

4.3.1 Analysis of the complexity

Intuitively, the optimal score is computed according to the following rules. In the first step ($i = 0$), no target words are generated and all positions in C are covered by the empty word e_0. In the next steps ($i > 0$), one additional target word is generated according to the most likely of two possibilities: i) sub-case $\phi_i > 0$, additional positions π_i are covered up to C, by producing a new center $\bar{\pi}$; ii) sub-case $\phi_i = 0$, no further positions are covered, hence the center remains unaltered.

The above formulation presents two features which make the search problem suitable to the application of DP [18]:

- *Optimal substructure*: i.e. the optimization problem contains within it optimal solutions to identical subproblems of smaller size. More specifically, optimal solutions for length i embed optimal solutions for length $i-1$.
- *Overlapping subproblems*: i.e. the number of subproblems to be solved is "relatively" small so that an algorithm typically revisits the same subprob-

lems over and over again. In particular, searching an optimal solution of length l which covers all source positions would require searching among all target strings of length l, all alignments between source and target positions (plus the e_0). This corresponds to a search space of size $(l+1)^m \, |\mathcal{E}|^l$. On the other hand, the number of subproblems $Q_i(\mathcal{C}, \bar{\pi}, e', e)$ to be solved is upper-bounded by the significantly lower number $(l+1) \, 2^m \, m \, |\mathcal{E}|^2$.

Each subproblem $Q_i(\mathcal{C}, \bar{\pi}, e', e)$ is solved through the maximization over generic $e \in \mathcal{E}$, $\emptyset \subseteq \pi_i \subseteq \mathcal{C}$ so that $\bar{\pi}_i = \bar{\pi}$, and $\bar{\pi}'$. By assuming a maximum ϕ_{max} for the fertilies values this maximization is performed over a set of partial hypotheses which is smaller than[7]

$$|\mathcal{E}| \, (\phi_{max}+1) \begin{pmatrix} m \\ \phi_{max} \end{pmatrix} (m+1)$$

and for each of those the number of operations is constant. Hence we can conclude that the complexity of the search algorithm is $O\left(2^m \, m^3 \, \phi_{max} \begin{pmatrix} m \\ \phi_{max} \end{pmatrix} |\mathcal{E}|^3\right)$, by constraining the output strings to be at most two times longer than the input length.

4.3.2 Search algorithm

A search algorithm exploiting dynamic programming is shown in Figure 4.3. The algorithm works iteratively by expanding, at each step i, a pool (set) of theories corresponding to partial solutions of length i. This results in a new pool of theories of length $i+1$, which is expanded at step $i+1$, and so on. At the begin, the best solution *best-th* is initialized with a fake complete solution (line 1-2), and the pool is initialized with theories generating the null word (lines 3-5). These theories cover source positions but do not generate target words. Theories *th* are characterized by the following three attributes: the state of the

[7] We assume that $\phi_{max} < \frac{m}{2}$.

theory *state*[*th*], which specifies the coverage set C, the center of the last cept $\bar{\pi}$, and the last two target words e' and e; the score of the theory *score*[*th*]; and the backpointer *bp*[*th*], which refers to the theory *th* originates from.

At each step i (lines 7-19), the pool of expanded theories ($pool[i+1]$) is initialized with the empty set (line 8). Hence, theories of the current set ($pool[i]$) that score better than the so-far best found complete solution (*best-th*) are considered for completion or expansion (lines 10-11). In fact, given that theory expansion always decreases the score, such theories are not worth to be expanded. Every surviving theory *th1* which covers all source positions (ISSOLUTION(*th1*)) returns true, line 11) is checked against the so far best complete solution *best-th*, which is eventually updated (line 12). Otherwise, *th1* is expanded by deriving all possible theories *th2* with one additional target word, (outcome of EXPANDTHEORY(*th1*), line 13). For each theory *th2*, recombination over $pool[i+1]$ is performed by checking the existence of other theories having the same state (line 14-17). As a result, for each state only the best scoring theory is maintained. After all theories in $pool[i]$ have been expanded, less promising theories in $pool[i+1]$ are pruned (PRUNETHEORIES). In particular, theories which score below the local optimum by a given threshold are eliminated, as well as theories out of the top N-best, where N is a given threshold. These criteria are applied, first, to all theories with a fixed coverage set, then to all theories of the pool. After theory pruning, the step variable i is increased by one. The iterative procedure stops once there are no more theories to expand, i.e. $pool[i]$ is empty. In this case, the optimal translation is recovered from *best-th* through its back-pointer, which was properly set at its creation time by procedure EXPANDTHEORY (line 20).

In order to reduce the possibly huge number of theories to generate, or equivalently sub-problems to solve, two methods are used:

- *Reordering constraint* (procedure EXPANDTHEORY): a smaller number of theories is generated by applying the so-called IBM constraint each time a

```
DP-SEARCH
 1  best-th ← DUMMYCOMPLETETHEORY(m)
 2  score[best-th] ← −∞
 3  pool[0] ← ∅
 4  for each th ∈ NULLWORDTHEORIES(m)
 5       do ADDTHEORY(pool[0], th)
 6  i ← 0
 7  while pool[i] ≠ ∅
 8       do pool[i+1] ← ∅
 9          for each th1 ∈ pool[i]
10               do if score[th1] > score[best-th]
11                    then if ISSOLUTION(th1)
12                              then best-th ← th1
13                    else  for each th2 ∈ EXPANDTHEORY(th1)
14                                do if found ← FINDTHEORY(pool[i+1], state[th2])
15                                     then if score[th2] > score[found]
16                                               then REPLACETHEORY(found, th2)
17                                     else  ADDTHEORY(pool[i+1], th2)
18          PRUNETHEORIES(pool[i+1])
19          i ← i+1
20  BACKTRACKSOLUTION(best-th)
```

Figure 4.3: DP-based search algorithm.

new source position is covered: only select one of the first 4 empty positions, from left to right.

- *Probability cutoff* (procedure EXPANDTHEORY): less target word hypotheses are considered for each source word by trimming the translation probabilities $p(f \mid e)$ at a given quantile and at a maximum number of entries, .99 and 15, respectively.

More details about these techniques and their impact on a search algorithm for Model 4 can be found in [79]. Finally, Figure 4.4 shows how theories are

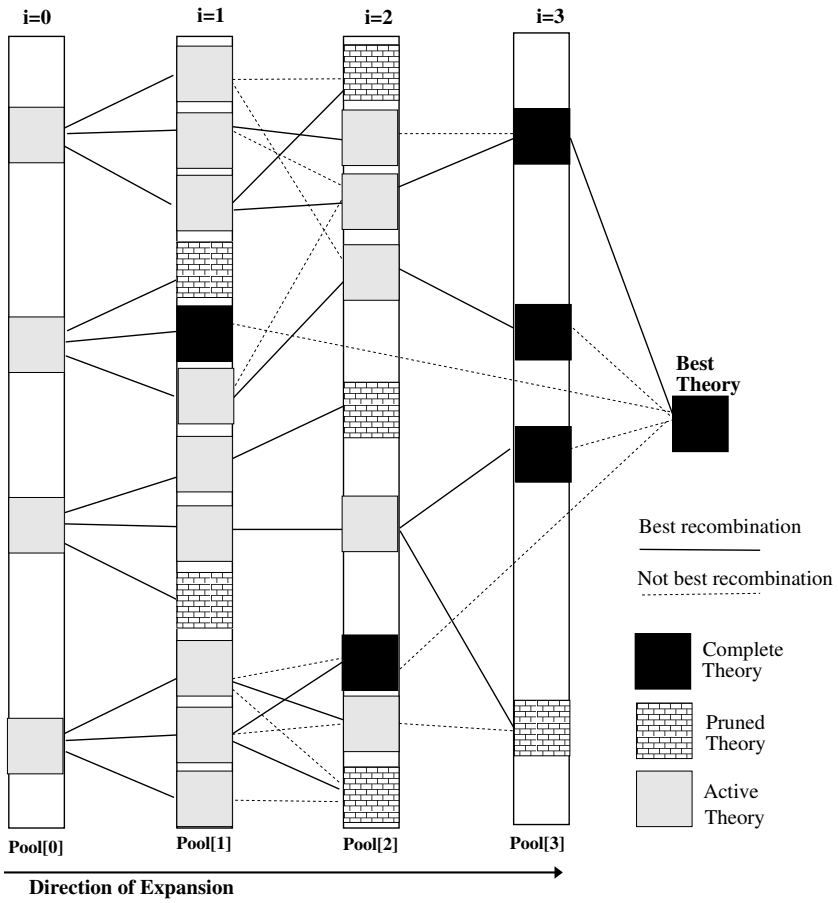

Figure 4.4: Expansion, recombination and pruning of theories during the search process.

generated, recombined and pruned during the search process.

4.3.3 Extension to word-graphs

The algorithm described in the previous Section looks for the best translation only and therefore will also call it 1-best decoder. As will be shown later in Section 5.3, it can be useful to generate more translation alternatives for each input sentence. A relatively easy modification of the 1-best decoder allows to produce a Word Graph (WG) (see Section A) representing all translation hypotheses considered during the search. This augmented decoder will be referred

to as WG decoder; similar work was proposed in [82]. Practically, instead of storing one backpointer to the best entering theory, we store all incoming theories by means of a list of backpointers. If we look at Figure 4.4 again, all edges entering in a theory are saved, included the dotted ones, which are removed by the 1-best decoder instead.

```
DP-SEARCH
 1  best-th ← DUMMYCOMPLETETHEORY(m)
 2  score[best-th] ← −∞
 3  pool[0] ← ∅
 4  for each th ∈ NULLWORDTHEORIES(m)
 5       do ADDTHEORY(pool[0],th)
 6  i ← 0
 7  while pool[i] ≠ ∅
 8       do pool[i+1] ← ∅
 9          for each th1 ∈ pool[i]
10               do if ISSOLUTION(th1)
11                    then ADDBACKPOINTERS(best-th,th1)
12                         if score[th1] > score[best-th]
13                           then REPLACETHEORY(best-th,th1)
14                    else  for each th2 ∈ EXPANDTHEORY(th1)
15                               do if found ← FINDTHEORY(pool[i+1],state[th2])
16                                    then ADDBACKPOINTERS(found,th2)
17                                         if score[th2] > score[found]
18                                           then REPLACETHEORY(found,th2)
19                                    else  ADDTHEORY(pool[i+1],th2)
20          PRUNETHEORIES(pool[i+1])
21          i ← i+1
22  WORDGRAPHGENERATION(best-th)
```

Figure 4.5: DP-based search algorithm for the generation of a word-graph.

The algorithm presented in Section 4.3.2 is modified as shown in Figure 4.5. First of all, the comparison between the scores of the actual theories and the best theory (Figure 4.3, line 10) is avoided because we are interested in all

translation hypotheses, and not only in the best one.
At every recombination step (lines 10-13 and 15-18), we first update the list of backpointers of the pre-existing theory (*best-th* and *found*, respectively) with the new theory (procedure ADDBACKPOINTERS), and then substitute the winner hypothesis in $pool[i+1]$ (procedure REPLACETHEORY).
Moreover, in the final step we do not recover the best translation anymore, but we generate a WG by backtrackifrom *best-th*. Pruning is possibleg as well, and all edges entering in a pruned theory can be removed.
Time for decoding only slighty increases due to backpointers storing; in fact, loser theories in the recombination steps are not removed, but they are not expanded. Time for searching the best translation is not more time costly, if we store also the best backpointer.
There is an increasing memory consumption because we store more backpointers, and because garbaging is less effective. Experiments show that the process requires about xx% more memory.

4.4 Phrase-based translation

The advantages of the statistical translation approach are advocated by the many papers on the subject, which followed its first introduction. Of course, there have been also attempts to overcome some of its shortcomings. In particular, statistical phrase-based translation models have recently emerged as a mean to cope with the limited use of context that Model 4 makes in order to guess word translations (lexicon model) and word positions (distortion model). Phrase models rely on statistics of phrase pairs, which can be automatically extracted from a word-aligned parallel corpus [41].
In the following, we briefly describe a known technique to extract phrase pairs, and then introduce a novel phrase-based translation framework which is tightly related to Model 4, so that the same search algorithm can be used. The same

search algorithm presented in Section 4.3.2 can be used because modifications are given only in the probability distribution and not in the generative process.

4.4.1 Phrase-pair extraction

The here used method exploits so called *union alignments* between sentence pairs of the training corpus [41]. Given strings $\mathbf{f} = f_1, \ldots, f_m$ and $\mathbf{e} = e_1, \ldots, e_l$, a direct alignment $\mathbf{a}$ (from $\mathbf{f}$ to $\mathbf{e}$) and an inverted alignment $\mathbf{b}$ (from $\mathbf{e}$ to $\mathbf{f}$), the union alignment is defined as:

$$\mathbf{c} = \{(j,i) : a_j = i \ \vee \ b_i = j\} \tag{4.36}$$

It is easy to verify that while $\mathbf{a}$ and $\mathbf{b}$ are many-to-one alignments, $\mathbf{c}$ can be a many-to-many alignment. Moreover, the union alignment does not necessarily cover all source and target positions (see the example in Figure 2).

Given a source-target sentence pair $(\mathbf{f}, \mathbf{e})$ and an union alignment $\mathbf{c}$, let J and I denote two closed intervals within the positions of $\mathbf{f}$ and $\mathbf{e}$, respectively. We say that I and J form a *phrase pair* under $\mathbf{c}$ if and only if $\mathbf{c}$ aligns all source positions J with target positions contained in I, and all target positions I with source positions contained in J. (See examples in the caption of Figure 2.). From the point of view of the direct alignment, phrases may include zero-fertility words, in the target, and words mapped in the null word, in the source. See, for instance, the phrases in Figure 2 which contain the target words `after` and `tomorrow` and the source word `beh`.

Given a parallel corpus provided with Viterbi alignments in both directions:

$$\left\{\left(\mathbf{f}^s, \mathbf{e}^s, \mathbf{a}^s, \mathbf{b}^s\right) : s = 1, \ldots, S\right\}$$

we can compute all phrase pairs occurring in it:

$$\mathcal{P} = \left\{\left(\tilde{f}^p, \tilde{e}^p\right) : p = 1, \ldots, P\right\} \tag{4.37}$$

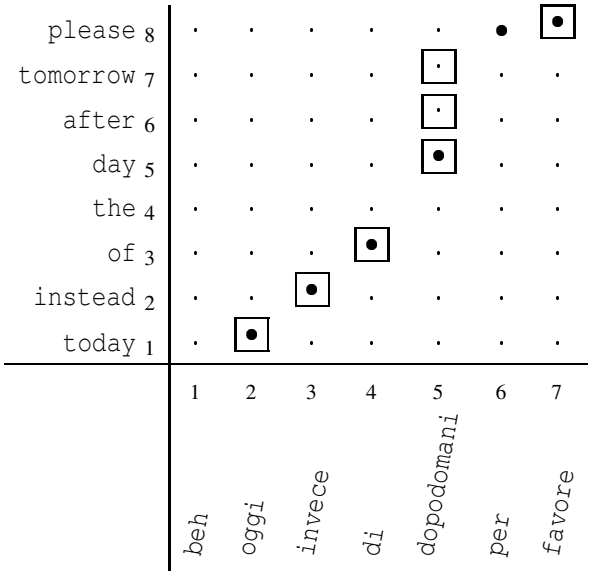

Figure 4.6: Examples of source phrase, target phrase, and overlapped direct (•) and inverted ($\boxed{\cdot}$) alignments. The union alignment corresponds to all points •, $\boxed{\cdot}$ and $\boxed{\bullet}$. Examples of resulting phrase pairs are: (beh#oggi#invece, today#instead), (di, of#the), (di#dopodomani, of#the #day#after#tomorrow).

Practically, in order to limit the number of phrases, the maximum length of I and J is limited to some value k. It is worth noticing that the set $\mathcal{P}$ also includes phrase pairs with one single target word.

4.4.2 Phrase-based translation framework

We assume that an augmented target vocabulary $\tilde{\mathcal{E}}$ is obtained from $\mathcal{E}$ by including all target phrases in $\mathcal{P}$. Hence, the search criterion (4.26) is modified as follows:

$$\tilde{\mathbf{e}}^* = \arg\max_{\tilde{\mathbf{e}}} \max_{\mathbf{a}} \Pr(\mathbf{f}, \mathbf{a} \mid \tilde{\mathbf{e}}) \Pr(\tilde{\mathbf{e}}) \tag{4.38}$$

where $\tilde{\mathbf{e}}$ ranges over all strings of $\tilde{\mathcal{E}}$.

Next subsection presents the phrase-based LM $\Pr(\tilde{\mathbf{e}})$, which extends a standard word-based LM, introduced in Section 4.2.1. Later, three phrase-based transla-

tion models $\Pr(\mathbf{f}, \mathbf{a} \mid \tilde{\mathbf{e}})$ are introduced, which maintain the same parametric form of Model 4. Depending on the model, fertility, lexicon, and distortion probabilities are estimated by assuming a sample $\mathcal{P}$ of phrases or available parameter estimates of Model 4, and the augmented target vocabulary $\tilde{\mathcal{E}}$.

4.4.3 Phrase-based language model

The language model probability of a string of phrases $\tilde{\mathbf{e}} = \tilde{e}_1, \ldots, \tilde{e}_l$, where $\tilde{e}_i = e_{i,1}\#..\#e_{i,k_i}$ $(i = 1, \ldots, l)$, is computed through the following decomposition:

$$\begin{aligned} \Pr(\tilde{\mathbf{e}}) &= p(l)\ p(\tilde{\mathbf{e}} \mid l) \\ &= p(l) \prod_{i=1}^{l} p(\tilde{e}_i \mid \tilde{e}_1, .., \tilde{e}_{i-1}) \\ &= p(l) \prod_{i=1}^{l} p(k_i) \prod_{h=1}^{k_i} p(e_{i,h} \mid \tilde{e}_1, .., \tilde{e}_{i-1}, e_{i,1}, \ldots, e_{i,h-1}) \end{aligned} \quad (4.39)$$

The first factor $p(l)$ is a probability distribution of the length l of a string of phrases (see Equation (4.6); the first product multiplies the phrase length probability of each phrase, which are taken uniform within the interval $[1..k]$; the last product is the word-based language model probability of the whole sequence of words in $\tilde{\mathbf{e}}$.

By considering a trigram LM and renumbering the phrase sequence $\tilde{\mathbf{e}} = e_{1,1}, .., e_{1,k_1}, \ldots, e_{l,1}, .., e_{l,k_l} = e_1, \ldots, e_n$, we can rewrite:

$$\Pr(\tilde{e}) = p(l) \prod_{i=1}^{l} p(k_i) \prod_{i=1}^{n} p(e_i \mid e_{i-2}, e_{i-1}) \quad (4.40)$$

where $n = \sum_{i=1}^{l} k_i$

Notice that the language model (4.39) is *deficient*, i.e. it assigns positive probabilities even to phrases outside the extended vocabulary.

4.4.4 Sample-based phrase model

This phrase model exploits a counting probability measure defined on the phrase sample $\mathcal{P}$. Hence, the relative frequency of a given phrase pair $(\tilde{f},\tilde{e})$ in the sample $\mathcal{P}$ is interpreted as the probability of the phrase-pair, given the training data. Hence, for any phrase-pair $(\tilde{f},\tilde{e})$ and fertility value ϕ, we define the following statistics over $\mathcal{P}$, which are used to compute basic probabilities of the translation model:

$$\begin{aligned} N(\tilde{f},\phi,\tilde{e}) &= \sum_{p=1}^{P} \delta(\tilde{f}^p=\tilde{f})\ \delta(\tilde{e}^p=\tilde{e})\ \delta(|\tilde{f}^p|=\phi) \\ N(\phi,\tilde{e}) &= \sum_{\tilde{f}} N(\tilde{f},\phi,\tilde{e}) \\ N(\tilde{e}) &= \sum_{\phi} N(\phi,\tilde{e}) \end{aligned}$$

where $\delta(\mathcal{R})=1$ if $\mathcal{R}$ is true and 0 otherwise.

Fertility Model. The fertility model is defined as the sample conditional frequency of a fertility value ϕ given a target phrase $\tilde{e}$:

$$\tilde{p}_S(\phi \mid \tilde{e}) = \frac{N(\phi,\tilde{e})}{N(\tilde{e})} \tag{4.41}$$

Lexicon Model. The lexicon model is defined as the sample conditional probability of a tablet τ given a fertility value ϕ and a target phrase $\tilde{e}$:

$$\tilde{p}_S(\tau \mid \phi,\tilde{e}) = \frac{N(\tilde{f}(\tau),\phi,\tilde{e})}{N(\phi,\tilde{e})} \tag{4.42}$$

where $\tilde{f}(\tau)$ trivially transforms τ into a phrase. The implicit assumption that the tablet must correspond to a source phrase, i.e. it must cover consecutive positions, is made explicit by the following distortion model.

Distortion Model. The sample-based distortion model assigns the first tablet position the same probability given by the Model 4 distortion model, but constrains successive positions to be adjacent. Hence, assuming that $\bar{\pi}$ is the center of the cept preceding π, we have:

$$\tilde{p}_S(\pi \mid \phi, \bar{\pi}) = p_{=1}(\pi_{\cdot,1} - \bar{\pi}) \prod_{k=2}^{\phi} \delta(\pi_{\cdot,k} - \pi_{\cdot,k-1} = 1) \tag{4.43}$$

Besides limiting possible permutations of a tablet, in accordance with the phrase extraction method, the sample-based phrase model strongly relies on statistics extracted from the training corpus. Given the significantly larger number of parameters, we should expect probability estimates with a lower bias, with respect to Model 4, but higher variance when data-sparseness increases.

4.4.5 Composition-based phrase model

This model is build-up from an existing word-based translation model and the augmented vocabulary. Given a target phrase $\tilde{e} = e_1\#\ldots\#e_d$, fertility and lexicon models are defined, which reduce to those of Model 4 in the case $d = 1$. The original distortion model is maintained, assuming that it does not depend on the target phrase.

Fertility Model. The probability of ϕ is computed by integrating over all possible fertilities $\phi_1, \ldots, \phi_d$ of the words in $\tilde{e}$, such that $\phi_1 + \ldots + \phi_d = \phi$, i.e.:

$$\tilde{p}_C(\phi \mid \tilde{e} = e_1\#\ldots\#e_d) = \begin{cases} \displaystyle\sum_{\phi_d=0}^{\phi} p(\phi_d \mid e_d)\, \tilde{p}_C(\phi - \phi_d \mid e_1\#\ldots\#e_{d-1}) & \text{if } d > 1 \\ p(\phi \mid e_d) & \text{otherwise} \end{cases} \tag{4.44}$$

where $p(\phi \mid e)$ denotes the Model 4 fertility probability. If ϕ_{max} is the maximum fertility associated with single words, the maximum fertility value for $\tilde{e}$ is

$d\ \phi_{max}$.

Lexicon Model. Given a fertility value ϕ and a phrase $\tilde{e}$, the probability of a tablet τ for $\tilde{e}$ is computed by integrating over all possible word alignments from tablet τ to phrase $\tilde{e}$, i.e.:

$$\tilde{p}_C(\tau \mid \phi, \tilde{e}) = d^{-\phi} \sum_{a_1=1}^{d} .. \sum_{a_\phi=1}^{d} p(\tau_{.,1} \mid e_{a_1}) .. p(\tau_{.,\phi} \mid e_{a_\phi}) \quad (4.45)$$

$$= \prod_{k=1}^{\phi} \left\{ \frac{1}{d} \sum_{i=1}^{d} p(\tau_{.,k} \mid e_i) \right\} \quad (4.46)$$

where the factor $d^{-\phi}$ corresponds to taking a uniform prior distribution over the alignments, and $p(\tau_{.,.} \mid e)$ is the Model 4 lexicon probability. The last formula, which is much faster to compute, results from a property of the sum, already shown in [12].

It is worth noticing that by inheriting the word-based distortion model, this phrase model does not meet the requirement of phrases to cover consecutive source positions. In fact, it will moderately favor true phrase-pairs, since the distortion model tends to reward permutations with no holes inside. Clearly, the composition-based model embeds less knowledge about the sample $\mathcal{P}$, i.e. only the set of target phrases, and permits the same word reordering freedom as Model 4. Given that lexicon and fertility probabilities rely on parameters estimated of simpler events, i.e. single word fertilities and translations, the composition-based model should feature more robustness with respect to data sparseness, but less precision (higher bias) with respect to the sample-based model.

Notice that in order to bound the decoding complexity with this model, for each target hypothesis $\tilde{e}$, the search algorithm will only considers tablets τ such that the pair $(f(\tau),\tilde{e})$ occurs in $\mathcal{P}$. As a consequece, there will be no difference between the theories generated with the composition- and sample-based models,

but the computed scores.

4.4.6 Interpolation-based phrase model

This model tries to exploit the complementarity of the previous two models, by linearly combining their components. The resulting fertility, lexicon, and distortion models, as well as the applied interpolation weights, are the following.

$$\begin{aligned}
\tilde{p}_I(\phi \mid \tilde{e}) &= \lambda(\tilde{e})\ \tilde{p}_S(\phi \mid \tilde{e})\ + \\
&\quad (1-\lambda(\tilde{e}))\ \tilde{p}_C(\phi \mid \tilde{e}) && (4.47)\\
\tilde{p}_I(\tau \mid \phi, \tilde{e}) &= \lambda(\phi, \tilde{e})\ \tilde{p}_S(\tau \mid \phi, \tilde{e})\ + \\
&\quad (1-\lambda(\phi, \tilde{e}))\ \tilde{p}_C(\tau \mid \phi, \tilde{e}) && (4.48)\\
\tilde{p}_I(\pi \mid \tau, \phi, \tilde{e}, \bar{\pi}) &= \lambda(\tau, \phi, \tilde{e})\ \tilde{p}_S(\pi \mid \phi, \bar{\pi})\ + \\
&\quad (1-\lambda(\tau, \phi, \tilde{e}))\ \tilde{p}_C(\pi \mid \phi, \bar{\pi}) && (4.49)
\end{aligned}$$

where

$$\lambda(\tilde{e}) = \frac{N(\tilde{e})}{N(\tilde{e}) + D(\tilde{e})} \qquad D(\tilde{e}) = \sum_{\phi} \delta(N(\phi, \tilde{e}) > 0)$$

$$\lambda(\phi, \tilde{e}) = \frac{N(\phi, \tilde{e})}{N(\phi, \tilde{e}) + D(\phi, \tilde{e})} \quad D(\phi, \tilde{e}) = \sum_{\tau} \delta(N(\tilde{f}(\tau), \phi, \tilde{e}) > 0)$$

$$\lambda(\tau, \phi, \tilde{e}) = \begin{cases} \sqrt[\phi-1]{.95} & \text{if } (\tilde{f}(\tau), \tilde{e}) \in \mathcal{P} \wedge \phi > 1 \\ 0 & \text{otherwise} \end{cases}$$

In particular, the model smoothes sample frequencies of the fertility and lexicon models by means of the well known method by [87]. Sample frequencies of the distortion model are instead smoothed according to the *a priori* probability that τ will cover adjacent positions, given that the pair $(\tilde{f}(\tau), \tilde{e})$ is observed in $\mathcal{P}$. Empirically we set this probability to .95 for a tablet of two elements, and let this probability grow as $\sqrt[\phi-1]{\cdot}$ for larger values of the fertility ϕ.

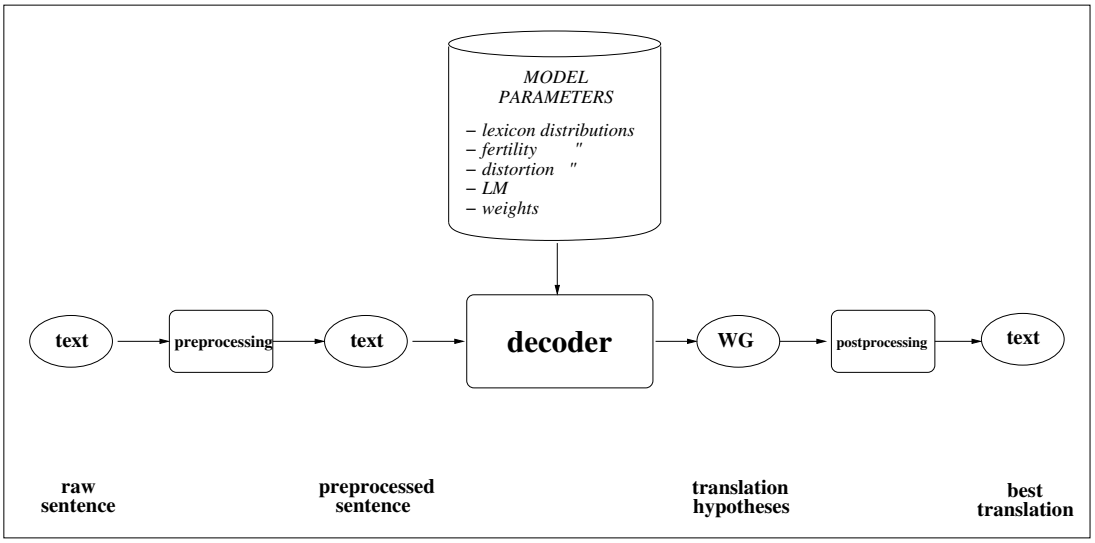

Figure 4.7: The ITC-irst Statistical Machine Translation system at run time: after preprocessing, the input sentence is sent to the decoder that, given the model parameters, search for the best hypothesis. A final postprocessing step provides the actual translation.

We expect the interpolation-based model to trade-off between the less precise but more stable parameters of the composition-based model and the sharper but less reliable estimates of the sample-based model.

4.5 The ITC-irst SMT system

The architecture of the Statistical Machine Translation system at run time is shown in Figure 4.7. The main module is the decoder, which implements the search algorithm presented in Section 4.3. Given a sentence in the source language, it provides as output a WG including alternative translation hypotheses in the target language, from which the best one or an N-best list can be extracted. Before and after the decoding phase, two additional steps are performed: first, the raw input sentence is preprocessed to normalize words and reduce data sparseness (see section 4.5.1 for details); finally, the best or the N-best hypotheses provided by the decoder are postprocessed in order to get the actual translation (section 4.5.2).

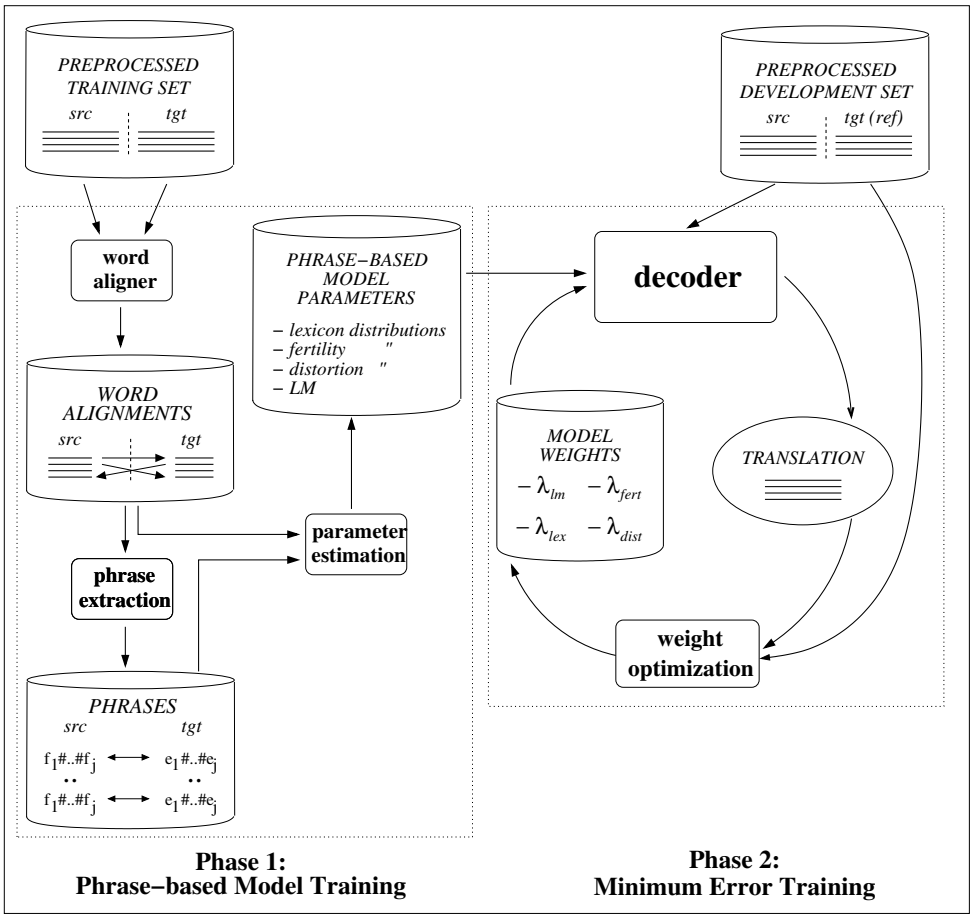

Figure 4.8: The two-phase architecture of the training system: first, the probability distributions of the phrase-based translation model are estimated by means of alignments (left side). Language model is estimated on a monolingual corpus. Then, the model weights of the submodels are computed by a minimum error training loop (right side).

Parameters of the statistical model are estimated by a dedicated training system whose two-phase architecture is shown in Figure 4.8. In the first phase the probability distributions of the phrase-based translation model are computed starting from a parallel training corpus. After preprocessing, word alignment with GIZA++ toolkit [59], and phrase extraction distributions of the word- and/or phrase-based model are estimated. Language model parameter is trained on a monolingual corpus, even larger than the parallel corpus.

In the second phase model weights can be optimized through a Minimum Er-

ror Training loop, which minimizes the translation errors made by the decoder running with a specific parameter set on a development corpus (see Section 5.3).

4.5.1 Preprocessing

Preprocessing of the data aims at normalizing words in order to reduce data sparseness. The same preprocessing steps are applied to both source and target sentences, obviously specialized for each language.

tokenization Words are separated from punctuation. Acronyms and abbreviations are also managed in this step.

number extraction Numbers are transformed into digits.

segmentation In Chinese, as well as in other Asian languages, there is no space between words. It is therefore necessary to separate the sequence of ideograms into words because the training and decoding algorithms assume sentences divided into words. The segmentation step for Chinese is described in [6]. Non-Asian languages like English do not need segmentation.

splitting As decoding complexity is more than linear in the sentence length, long sentences are tentatively split into shorter portions according to strong punctuation.

labeling Expressions belonging to a specific group of semantic classes are substituted with appropriate labels that encode class name and the value represented by the expression. Currently recognized classes include cardinal and ordinal numbers, week-day and month names, month days, years and percentages. This step is performed only for Chinese.

case normalization All the words are put in lowercase.

punctuation reinforcement Each sentence is enriched by inserting at its beginning the strong punctuation character with which the sentence ends (namely full stop, question or exclamation mark). This extra information helps the system to match correctly words and their order with respect to the sentence type (statement, question or exclamation).

4.5.2 Postprocessing

The best hypothesis found by the decoder in the target language need to be refined before be presented as the final translation. The following postprocessing steps are then applied:

unsplitting In the processing, long source sentences were split into shorter portions. The unsplitting step recovers the original situation, by merging together the target sentences being the translation of the portions belonging to the same source sentence.

phrases decomposition The sequence of words composing a phrase is decomposed into the single words.

label value instantiation Labels with abstract values are translated into the target language according to the label class and value.

number normalization Number in digits are substituted with their string representation.

case restoration At this point all the words are in lowercase: this step tries to restore the uppercase words by means of a Maximum Entropy tagger. In addition simple syntactical rules are applied (e.g. words following strong punctuation are put in upper case).

4.6 Experiments

We present experimental results on two translation tasks: from Chinese to English and from Italian to English. Both tasks consists of translating written sentences commonly used in tourism domain, which can be found in the so-called phrase books.
Evaluation regards the comparison of the three phrase-based models and the word-based model, their performance according with different amount of training data and different settings of search parameters. `BLEU`, `NIST` and `MWER` [8], described in Appendix B.2, are used to measure translation quality, while the average number of hypotheses, `Avg#Th`, gives an estimate of the computational effort, because it strictly correlates the decoding time as shown in Appendix B.2.

4.6.1 BTEC corpus

The Basic Travel Expression Corpus (BTEC) is a multi-lingual parallel corpus, originally created by ATR [77], and now jointly developed within the C-STAR consortium[9]. The corpus originates from a large collection of Japanese-English phrase-books. Starting from the English version, translations were added in Chinese and Italian[10]. Figure 4.9) shows an excerpt of BTEC. In general, according to the target language one or more translations were produced for each sentence to solve possible translation ambiguities.

4.6.2 Training and testing data

Two parallel corpora are extracted from BTEC for training purposes. In Table 4.1 detailed statistics on the two translation tasks are reported. It is worth

[8] The MT scoring tool was kindly provided by USC/ISI.

[9] www.c-star.org

[10] BTEC also contains, or plans to add, translations in Korean, German, French and Spanish are in progress. Since 2004, parts of the corpus have been made available to research groups participating in evaluation campaigns organized by the C-STAR partners.

请给我来些果汁。
Vorrei del succo di frutta, per favore.
I'd like some fruit juice, please.

我不停咳嗽，疼得厉害。
Non riesco a smettere di tossire e fa davvero male.
I can't stop coughing and it hurts really bad.

希望你们旅途愉快，谢谢。
Spero che abbia fatto un buon volo. Grazie.
Hope you have a pleasant flight. Thank you.

Figure 4.9: Sample of the Basic Travel Expression Corpus.

noticing that the Italian-English training data includes translation alternatives.

For testing purposes, we employed two data sets both Chinese-English and Chinese-English tasks. The first test (Q1) set consists of 3006 sentences aligned in Chinese, Italian and English. For each test sentence, only one reference is available. This test set is large enough to compute relatively sharp 95% confidence intervals [93], and also to test if differences in performance between two systems are statistically significant or not (see Appendix B.2). In this case, we applied a test of equality of the means of two normal distributions, and considered a confidence level α=0.05. Moreover, we considered a second test set (Q2) with only 506 sentences, each having 16 translation alternatives. This test set was used in the 2003 C-STAR evaluation campaign.

Table 4.2 reports detailed statistics about the two test sets for Chinese, Italian, and English. Figures related to the target language refer to the gold reference.

In both tasks, source and target texts were preprocessed as described in Section 4.5.1.

As described in Section 4.4, estimation of phrase-based models needs the availability of bidirectional alignments and estimates of Model 4, which were computed with the GIZA++ toolkit [60]. Translation phrase-pairs were extracted from both training corpora according to the method described in Section 5.1.

The number of extracted phrase-pairs (see Table 4.1) was 1.5M and 1.1M for Chinese-English and Italian-English, respectively.

		source		target		phrase
	sentences	vocabulary	words	vocabulary	words	pairs
Chinese-English	159K	15.4K	1120K	13.2K	1141K	1459K
Italian-English	52K	15.7K	451K	10.8K	480K	1074K

Table 4.1: Statistics of the training corpora for the Chinese-English and Italian-English tasks: number of sentences, number of words, vocabulary size, and number of extracted phrase pairs.

test set		source			target		
		sent.	words	OOV rate	words	OOV	PP
Q1	Chinese-English	3006	27117	3.7%	28782	3.3%	99
	Italian-English	”	28332	4.3%	”	4.4%	110
Q2	Chinese-English	506	3765	2.8%	3670	2.1%	84
	Italian-English	”	3509	4.1%	”	4.3%	94

Table 4.2: Statistics of the two test sets for the Chinese-English and Italian-English tasks: number of sentences, number of words and percentage of out-of-vocabulary words in the source sentences, number of words and perplexity of the gold target reference.

4.6.3 Comparison of translation models

The first experiment measures the performance of the three phrase-based translation models and Model 4 according to all considered scores. Tables 4.3 and 4.4 report results on the Chinese-English and Italian-English tasks, respectively; performance in BLEU, NIST, and WER with corresponding confidence intervals, are given for the test set Q1. For the sake of comparison, the same pruning parameter setting of the search algorithm was used for all phrase models, while different pruning settings were applied to Model 4. Different parameter settings of Model 4 are reflected by increasing average numbers of generated theories (column Avg#Th).

	BLEU	NIST	MWER	Avg#Th
wM4	15.8 (15.1-16.5)	4.56 (4.47-4.65)	72.1 (70.3-73.8)	93K
“	16.7 (15.9-17.4)	4.71 (4.63-4.81)	69.9 (68.1-71.6)	144K
“	17.1 (16.4-17.8)	4.80 (4.71-4.89)	68.8 (67.0-70.4)	181K
“	17.1 (16.4-17.9)	4.84 (4.75-4.92)	68.3 (66.6-69.9)	204K
pMs	20.5 (19.7-21.3)	5.04 (4.94-5.15)	69.6 (67.6-71.5)	193K
pMc	17.4 (16.6-18.0)	4.70 (4.61-4.78)	72.6 (70.7-74.5)	195K
pMi	19.7 (19.0-20.5)	5.00 (4.90-5.09)	70.0 (68.0-71.8)	222K

Table 4.3: Comparison of different translation models on the test Q1 in the Chinese-English task.

	BLEU	NIST	MWER	Avg#Th
wM4	34.9 (33.8-35.8)	6.94 (6.83-7.04)	46.0 (44.8-47.2)	49K
“	38.7 (37.6-39.6)	7.39 (7.28-7.50)	42.7 (41.5-43.8)	76K
“	39.4 (38.4-40.4)	7.49 (7.37-7.59)	42.0 (40.9-43.2)	106K
pMs	44.8 (43.7-45.8)	7.92 (7.80-8.03)	38.4 (37.2-39.7)	93K
pMc	38.5 (37.5-39.4)	7.26 (7.14-7.37)	43.5 (42.3-44.7)	95K
pMi	42.4 (41.4-43.4)	7.68 (7.56-7.79)	40.0 (38.8-41.2)	110K

Table 4.4: Comparison of different translation models on the test Q1 in the Italian-English task.

BLEU and NIST scores indicate that the sample-based (pMs) and the interpolation-based (pMi) models outperform Model 4 (wM4), in both translation tasks. For both scores and tasks, differences are statistically significant at level α=0.05. At more or less comparable computational loads, improvements in BLEU score range between 13% (I-E BLEU 39.4 vs. 44.8) to 19% (C-E BLEU 17.1 vs 20.5).

The composition-based model is comparable with Model 4; in fact, the ranking of the two models depends on the task and on the considered scores. Among the phrase models, the sample-based model (pMs) performs significantly better (α=0.05) than the composition-based model (pMc), and similarly to the interpolation-based model (pMi).

In the Chinese-English task only, a smaller MWER score is reported in favor of

	BLEU	NIST	MWER
wM4	30.02	7.10	57.53
“	31.57	7.25	54.56
“	32.28	7.31	53.28
“	32.29	7.28	52.7
pMs	34.43	7.43	56.63
pMc	32.11	7.22	58.48
pMi	33.79	7.48	56.74

Table 4.5: Comparison of different translation models on the test set Q2 in the Chinese-English task.

	BLEU	NIST	MWER
wM4	45.78	8.45	40.11
“	48.52	8.58	37.52
“	48.73	8.54	37.17
pMs	59.37	9.82	32.23
pMc	51.32	9.05	37.14
pMi	57.32	9.63	33.22

Table 4.6: Comparison of different translation models on the test set Q2 in Italian-English task.

Model 4, under pruning parameters generating comparable numbers of theories.

These observation are fully confirmed by results on the test set Q2, reported in Tables 4.5 and 4.6.

From a qualitative point of view, evident improvements by phrase-based models are in the lexical choice, local word reordering, and fluency of the output. For instance, in the first example of Table 4.7, the phrase-based model was able to correctly reorder to common multi-word expression `il cancello d' imbarco` (`the boarding gate`). In the second example, the phrase-based model produces a more fluent output by translating the preposition `in` and the adjective `quell'` jointly with their left and right contexts, respectively.

Source (Italian)	Dove è il cancello d' imbarco ?
Translation with wM4	Where is the gate before boarding ?
Translation with pMs	Where is the#boarding#gate ?
Source (Italian)	C' è un salone di bellezza in quell' albergo ?
Translation with wM4	Is there a beauty salon in the of that hotel ?
Translation with pMs	Is#there a beauty#salon#in that#hotel ?

Table 4.7: Examples of translations produced by word- and phrase-based models.

4.6.4 Incremental training data

In order to investigate the model behavior under different data-sparseness conditions, subsets of the training corpora of increasing size were extracted. BLEU, NIST and WER scores achieved by all models after training on each sub-corpus are plotted in Figure 4.10 and 4.11 for the Chinese-English and Italian-English tasks, respectively. A logarithmic scale is used for the x-axis.

In both translation tasks, the composition-based model performs very close to or even better than the sample-based model in the worst data-sparseness conditions (up to 20K and up to 5K training sentences for Chinese-English and Italian-English, respectively). In the Chinese-English task, the sample-based model gives NIST scores even worse than Model 4 up to 5K training sentence. BLEU scores reward a bit more the sample-based phrase model, but in general the same conclusion can be drawn: among the considered models, this is the most sensitive to the amount of training data.

Remarkably, the interpolation-based model shows either the best or very close to the best scores, across all tasks and training conditions. These results seem to confirm the speculations made in the previous section.

4.6.5 Impact of zero fertility words

Figures 4.12 and 4.13 report BLEU and NIST scores of Model 4 and phrase models by varying the maximum number of zero-fertility words allowed in the target

string. This parameter is expressed in terms of a percentage of the length of the source string. Plots show that decreasing the possibility of insertions up to 5% improves BLEU performance for all phrase models, in particular for the sample-based model. NIST scores show lower improvements for phrase models and a small decrease for Model 4 are observed.
If the insertion of zero-fertility words is completely forbidden, a relative improvement of the BLEU score between 6% and 17% is observed for phrase-based models. On the contrary, Model 4 significantly worsens performance, reducing the BLEU score by a relative 8%-13%.
A rationale for the better performance yield by phrase models after limiting or forbidding target word insertions is that target phrases already contain extra words, as underlined in subsection 5.1. Moreover, an additional positive side-effect is the significantly lower number of generated theories, and, consequently, the reduced translation time.
In accordance with these results, subsequent experiments were carried out with the interpolated phrase-based model, and by inhibiting the insertion of zero-fertility words.

4.6.6 Length of source phrases

A factor that impacts on memory consumption of the decoding algorithm is the considered number of phrase pairs, which is strictly related to the size of the training corpus and the maximum allowed phrase length k. An experiment similar to that described in [41] was carried out to verify how translation accuracy varies with respect to the maximum allowed length of phrases. Results for both translation tasks by using all the available training data are shown in Tables 4.8 and 4.9.
Optimal or close to optimal performance is achieved with phrase length up to $k = 5$. However, a very good trade-off is reached for $k = 3$, which gives a slightly worse performance (less that 4% relative), but a significant reduction of

k	# phrases	BLEU	NIST	MWER	Avg#Th
1	88K	11.6 (11.0-12.3)	4.24 (4.15-4.33)	69.1 (67.5-70.7)	30K
2	174K	16.0 (15.3-16.7)	4.70 (4.61-4.78)	69.2 (67.5-70.7)	43K
3	335K	16.6 (15.9-17.3)	4.75 (4.67-4.83)	69.1 (67.4-70.7)	43K
4	534K	16.4 (15.8-17.1)	4.73 (4.64-4.81)	69.3 (67.6-70.8)	44K
5	745K	16.5 (15.8-17.2)	4.73 (4.65-4.82)	69.2 (67.6-70.8)	44K
6	951K	16.5 (15.8-17.2)	4.73 (4.64-4.82)	69.2 (67.6-70.8)	44K
7	1132K	16.5 (15.8-17.2)	4.73 (4.65-4.82)	69.2 (67.6-70.8)	44K

Table 4.8: Performance of the interpolation-based phrase model with different maximum phrase length k on the Chinese-English task.

k	# phrases	BLEU	NIST	MWER	Avg#Th
1	46K	28.12 (27.26-28.97)	6.61 (6.52-6.70)	49.5 (48.3-50.6)	16K
2	124K	41.68 (40.64-42.66)	7.93 (7.82-8.02)	38.8 (37.7-39.9)	45K
3	256K	45.88 (44.80-46.79)	8.22 (8.11-8.32)	35.9 (34.9-37.0)	55K
4	413K	46.35 (45.26-47.23)	8.24 (8.13-8.35)	35.5 (34.5-36.6)	57K
5	573K	47.03 (45.92-47.93)	8.30 (8.19-8.40)	35.0 (33.9-36.1)	58K
6	721K	47.36 (46.25-48.29)	8.34 (8.22-8.44)	34.8 (33.8-35.9)	58K
7	845K	47.45 (46.33-48.38)	8.34 (8.23-8.45)	34.7 (33.7-35.9)	58K

Table 4.9: Performance of the interpolation-based phrase model with different maximum phrase length k on the Italian-English task.

the number of phrases to store (46% for C-E, and 49% for I-E). In general, the introduction of phrases longer than 5 does not seem to pay off, probably due to the much lower chance of finding a match with the input.

From the computational point of view, the search algorithm reaches a maximum average number of generated theories around values $k = 4$ and $k = 5$, in both translation tasks.

These experiments were performed with the WG-decoder presented in Section 4.3.3. Interestingly, if we use 1-best decoder (see Section 4.3.2, the search effort slightly reduces and becomes stable for lengths above $k = 4$. An explanation could be that by exploiting longer phrases, the search algorithm is able

to find in fewer steps a complete solution which improves the initial dummy theory (see line 10 in Figure 4.3). It is well known that the efficiency of a search algorithm increases as long as a good candidate for a complete solution is available.

Again, phrases longer than 5 do not seem to impact on the search time, probably because of the little chance of finding long tablets, within the source string, whose corresponding phrases are in $\mathcal{P}$.

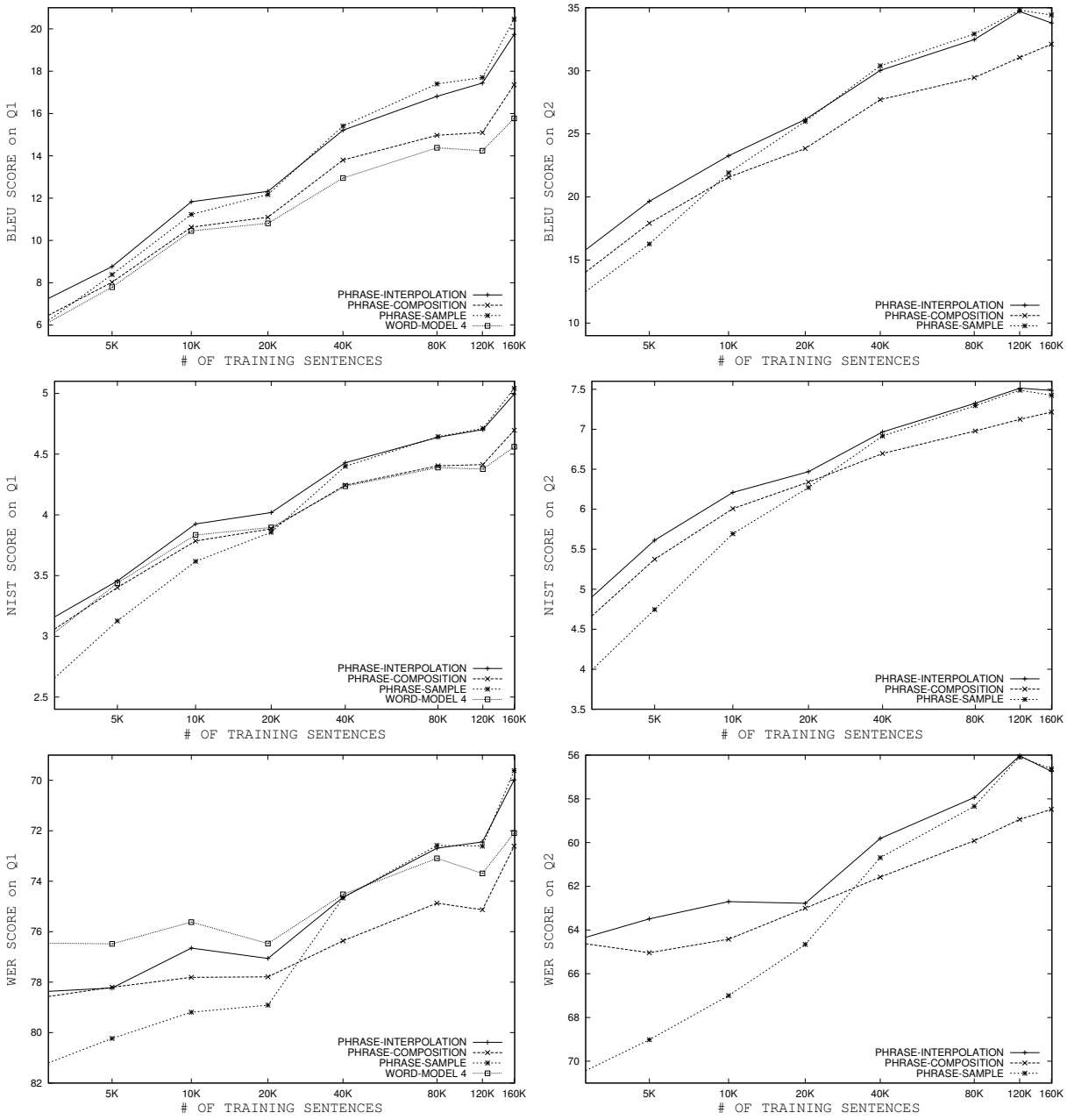

Figure 4.10: Performance of word- and phrase-based models vs. amount of training data on the Chinese-English task with respect different evaluation measures. Plots on the left refer to performance on test set Q2, while those on the right to test set Q2.

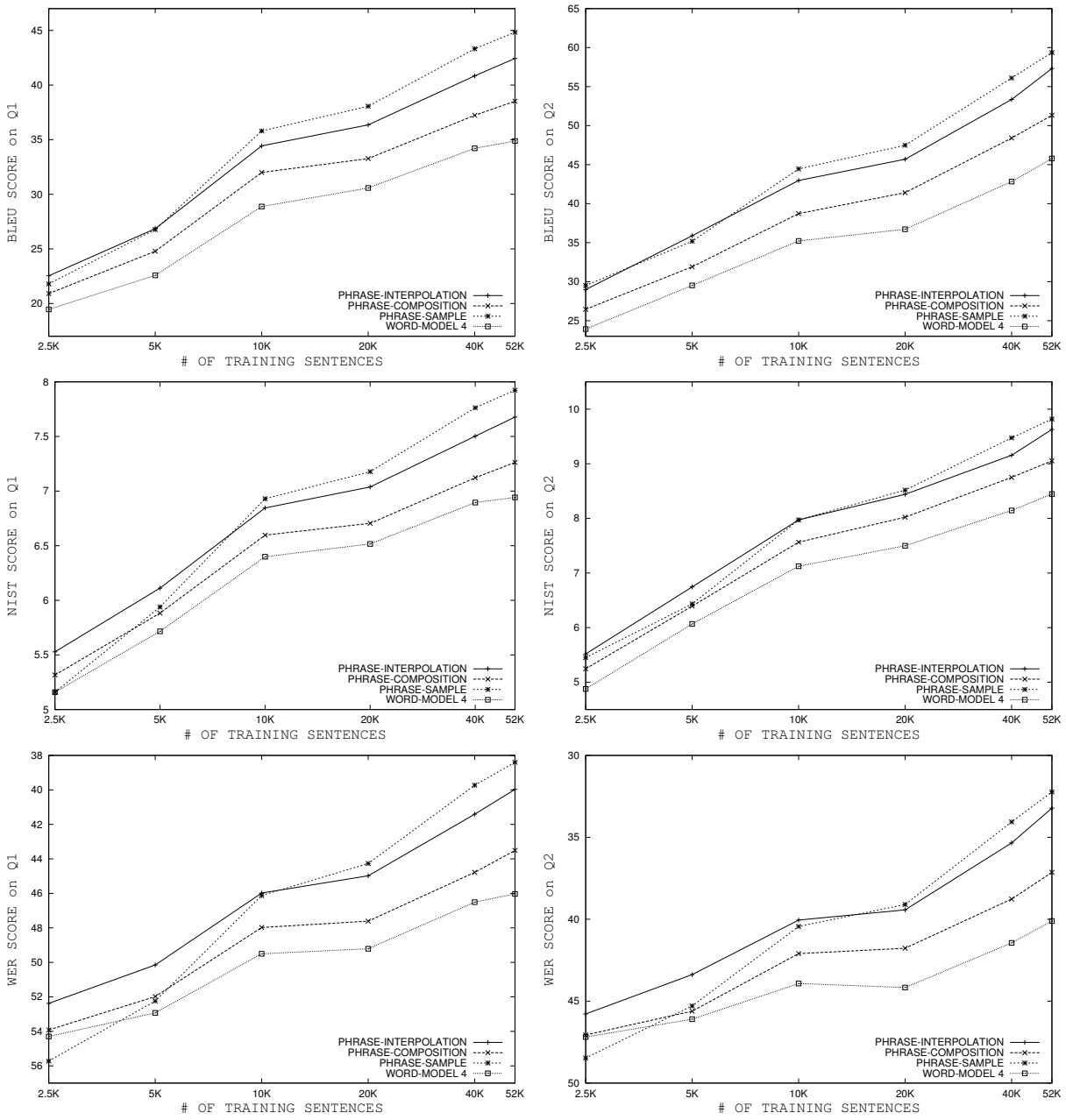

Figure 4.11: Performance of word- and phrase-based models vs. amount of training data on the Italian-English task with respect different evaluation measures. Plots on the left refer to performance on test set Q2, while those on the right to test set Q2.

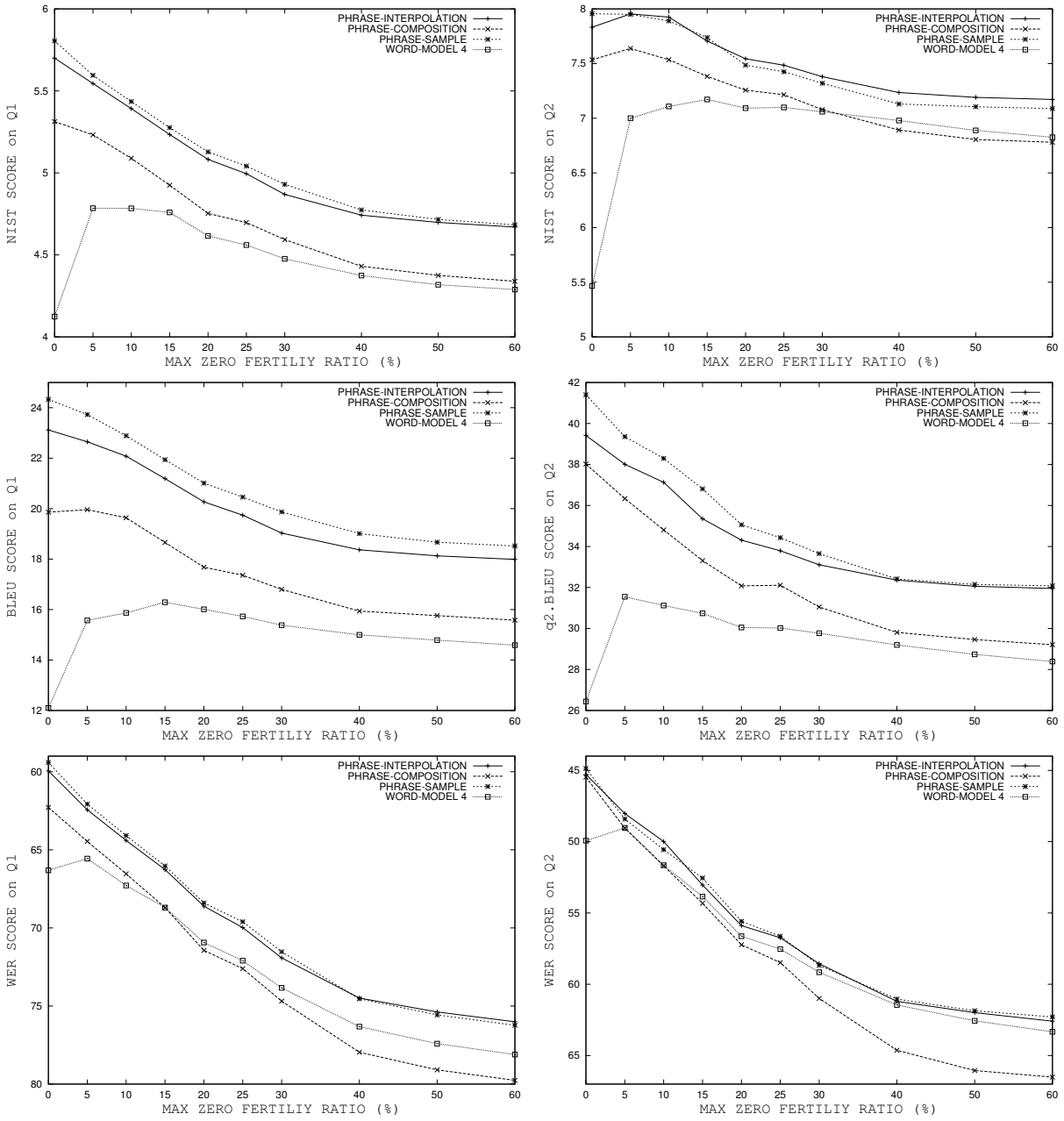

Figure 4.12: Performance of word- and phrase-based models vs. number of admitted zero-fertility words on the Chinese-English task.

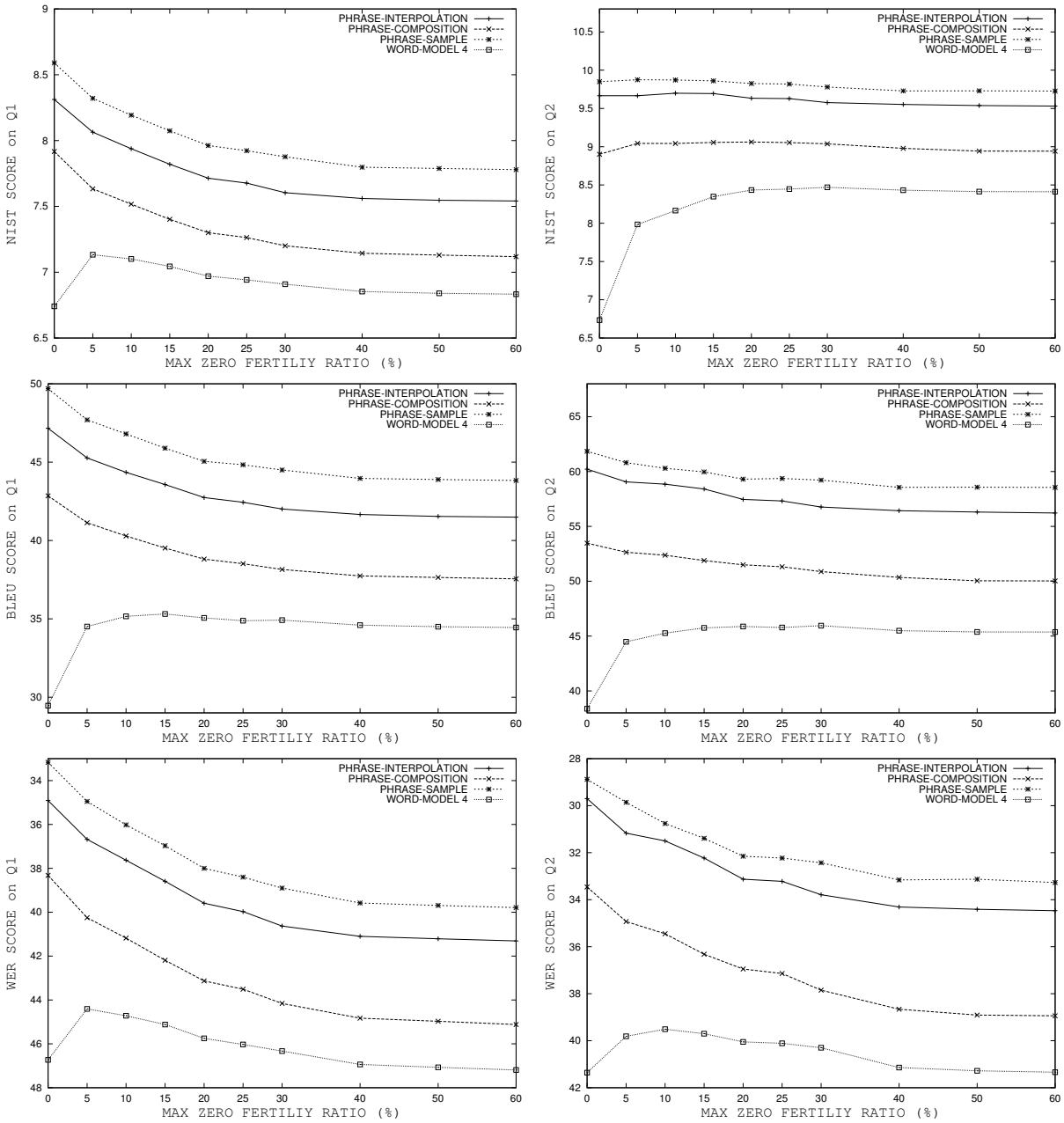

Figure 4.13: Performance of word- and phrase-based models vs. number of admitted zero-fertility words on the Italian-English task.

Chapter 5

Log-linear models for Statistical Machine Translation

In Chapter 4, we described the translation process from the point of view of the historically-leading source-channel framework, but we anticipated some of its drawbacks. SMT models based on the source-channel approach require an exact factorization of the involved probability distributions, and this often requires strong independence assumptions. For the same reason, the extension of a baseline statistical model by including additional dependencies may be very hard. Moreover, language and translation models should be "weighted" differently in order to achieve best performance.

These problems can be overcome by considering log-linear models instead, which can be formally derived within the Maximum Entropy framework [4] as shown in [39]. Introduced in ASR by Philips laboratories in the late '90s [9, 10], first attempts to apply these models to SMT are very recent [63, 58, 60, 92].

Next Section briefly describes a generic log-linear model for SMT and highlights its advantages over the source-channel based SMT model. Section 5.2 shows that the source-channel based SMT model is equivalent to a log-linear model. Finally, in Section 5.3.1 a procedure for training model parameters is described which is oriented to the minimization of the translation errors.

5.1 The SMT log-linear model

The description of an SMT model based on the Maximum Entropy framework assumes a generative process which is inverted with respect to that of the source-channel approach (see Section 4.2.4). Starting from an input string $\mathbf{f} = f_1, \ldots, f_m$ in the source language, a translation hypothesis of length l, $\tilde{\mathbf{e}} = \tilde{e}_0, \ldots, \tilde{e}_l$, is generated incrementally: at each step $i = 0, \ldots, l$ a new phrase $\tilde{e}_i$ is added and, possibly, some words of $\mathbf{f}$, not yet covered, are mapped to $\tilde{e}_i$.
Any triple $(\tilde{\mathbf{e}}, \mathbf{f}, \mathbf{a})$, or its shorthand $\mathbf{s}$, corresponds to a solution of length l obtained through the generative process. For any solution $\mathbf{s}$ of length l, we denote by $(\mathbf{s}, i)$ the portion of $\mathbf{s}$ built during steps from 0 to i ($i \leq l$).
It is worth remarking that any solution $\tilde{\mathbf{e}}$, partial or complete, can be generated from $\mathbf{f}$ in many different ways, each one corresponding to a specific alignment $\mathbf{a}$. The set of all these alignments is denoted with $\mathcal{A}(\mathbf{f}, \tilde{\mathbf{e}})$.

5.1.1 Alignment-based log-linear model

Given an input string $\mathbf{f}$, the best translation $\tilde{\mathbf{e}}^*$ of $\mathbf{f}$ is searched among all output strings through the following criterion:

$$\tilde{\mathbf{e}}^* = \arg\max_{\tilde{\mathbf{e}}} \Pr(\tilde{\mathbf{e}} \mid \mathbf{f}) \tag{5.1}$$

$$= \arg\max_{\tilde{\mathbf{e}}} \sum_{\mathbf{a} \in \mathcal{A}(\mathbf{f}, \tilde{\mathbf{e}})} \Pr(\tilde{\mathbf{e}}, \mathbf{a} \mid \mathbf{f}) \tag{5.2}$$

The summation over the alignments $\mathbf{a} \in \mathcal{A}(\mathbf{f}, \tilde{\mathbf{e}})$, which is considered as a hidden variable, introduces the freedom of generating $\tilde{\mathbf{e}}$ from $\mathbf{f}$ in many different ways. The framework of Maximum Entropy [4] provides a mean to directly estimate the posterior probability $\Pr(\tilde{\mathbf{e}}, \mathbf{a} \mid \mathbf{f})$, instead of using the Bayes decomposition into a language model $\Pr(\mathbf{e})$ and a translation model $\Pr(\mathbf{f} \mid \mathbf{e})$ introduced by the source-channel framework. This is determined through suitable real valued *feature functions* $h_r(\tilde{\mathbf{e}}, \mathbf{f}, \mathbf{a})$ and real parameters λ_r, $r = 1 \ldots M$, and takes the

parametric form:

$$\Pr(\tilde{\mathbf{e}}, \mathbf{a} \mid \mathbf{f}) = p_{\lambda}(\tilde{\mathbf{e}}, \mathbf{a} \mid \mathbf{f}) = \frac{\exp\left\{\sum_{r=1}^{M} \lambda_r h_r(\tilde{\mathbf{e}}, \mathbf{f}, \mathbf{a})\right\}}{\sum_{\tilde{\mathbf{e}}'} \sum_{\mathbf{a}' \in \mathcal{A}(\mathbf{f}, \tilde{\mathbf{e}}')} \exp\left\{\sum_{r=1}^{M} \lambda_r h_r(\tilde{\mathbf{e}}', \mathbf{f}, \mathbf{a}')\right\}} \quad (5.3)$$

where the denominator is needed for the sake of normalization.
By assuming the following notation:

$$\lambda = \{\lambda_1, \ldots, \lambda_M\} \quad (5.4)$$

$$R(\tilde{\mathbf{e}}, \mathbf{f}, \mathbf{a}; \lambda) = \sum_{r=1}^{M} \lambda_r h_r(\tilde{\mathbf{e}}, \mathbf{f}, \mathbf{a}) \quad (5.5)$$

and by exploiting (5.3), the previous search criterion (5.2) can be rewritten as follows:

$$\tilde{\mathbf{e}}^* = \arg\max_{\tilde{\mathbf{e}}} \sum_{\mathbf{a} \in \mathcal{A}(\mathbf{f}, \tilde{\mathbf{e}})} \Pr(\tilde{\mathbf{e}}, \mathbf{a} \mid \mathbf{f}) \quad (5.6)$$

$$= \arg\max_{\tilde{\mathbf{e}}} \sum_{\mathbf{a} \in \mathcal{A}(\mathbf{f}, \tilde{\mathbf{e}})} \frac{\exp\{R(\tilde{\mathbf{e}}, \mathbf{f}, \mathbf{a}; \lambda)\}}{\sum_{\tilde{\mathbf{e}}'} \sum_{\mathbf{a}' \in \mathcal{A}(\mathbf{f}, \tilde{\mathbf{e}}')} \exp\{R(\tilde{\mathbf{e}}', \mathbf{f}, \mathbf{a}'; \lambda)\}} \quad (5.7)$$

$$= \arg\max_{\tilde{\mathbf{e}}} \sum_{\mathbf{a} \in \mathcal{A}(\mathbf{f}, \tilde{\mathbf{e}})} \exp\{R(\tilde{\mathbf{e}}, \mathbf{f}, \mathbf{a}; \lambda)\} \quad (5.8)$$

because of the constancy of the denominator with respect to the maximization variable $\tilde{\mathbf{e}}$ in (5.7). Thus, the time-consuming renormalization can be avoided during the search.
For the sake of an efficient computation, the summation over the alignments $\mathbf{a}$ in (5.8) is approximated by the maximum operation, as usual. Finally, thanks to monotonicity of the exponential function, we end up with the criterion:

$$\tilde{\mathbf{e}}^* = \arg\max_{\tilde{\mathbf{e}}} \max_{\mathbf{a} \in \mathcal{A}(\mathbf{f}, \tilde{\mathbf{e}})} R(\tilde{\mathbf{e}}, \mathbf{f}, \mathbf{a}; \lambda) \quad (5.9)$$

5.1.2 The search algorithm for a log-linear model

The search criterion (5.9) can be detailed more precisely if the feature functions $h_1^M(\mathbf{s})$ can be decomposed in accordance with the steps of the generative process.

Let us assume that the score $R(\mathbf{s};\lambda) = R(\tilde{\mathbf{e}},\mathbf{f},\mathbf{a};\lambda)$ and the features $h_1^M(\mathbf{s}) = h_1^M(\tilde{\mathbf{e}},\mathbf{f},\mathbf{a})$ can be computed in terms of the following partial functions:

$$h_r(\mathbf{s};i) \qquad \forall\ r = 1,\ldots,M \tag{5.10}$$

$$R(\mathbf{s};\lambda,i) \;=\; \sum_{r=1}^{M} \lambda_r h_r(\mathbf{s};i) \tag{5.11}$$

where $R(\mathbf{s};\lambda,i)$ represents the cost of the i-th step of the generative process of $\mathbf{s}$, and $h_r(\mathbf{s};i)$ the contribution of the r-th feature to this cost. It is worth remarking that the computation of $R(\mathbf{s};\lambda,i)$ and $h_r(\mathbf{s};i)$ depends on the partial solution $(\mathbf{s},i)$. It trivially results that:

$$h_r(\mathbf{s}) \;=\; \sum_{i=0}^{l} h_r(\mathbf{s};i) \qquad\qquad \forall\ r = 1,\ldots,M \tag{5.12}$$

$$\begin{aligned} R(\mathbf{s};\lambda) \;&=\; \sum_{r=1}^{M} \lambda_r h_r(\mathbf{s}) = \sum_{r=1}^{M} \lambda_r \sum_{i=0}^{l} h_r(\mathbf{s};i) = \\ &=\; \sum_{i=0}^{l}\sum_{r=1}^{M} \lambda_r h_r(\mathbf{s};i) = \sum_{i=0}^{l} R(\mathbf{s};\lambda,i) \end{aligned} \tag{5.13}$$

Hence, the search criterion (5.8) becomes:

$$\mathbf{s}^* \;\approx\; \arg\max_{\mathbf{s}} R(\mathbf{s};\lambda) \tag{5.14}$$

$$\;=\; \arg\max_{l}\max_{\mathbf{s}} \sum_{i=0}^{l} R(\mathbf{s};\lambda,i) \tag{5.15}$$

and the best English translation $\tilde{\mathbf{e}}^*$ is extracted from the best solution $\mathbf{s}^* = (\tilde{\mathbf{e}}^*,\mathbf{f},\mathbf{a}^*)$.

It is important to stress here that each iteration of steps (iv.)-(viii.) of the generative process adds a new phrase $\tilde{e}$ to the target string; hence, a partial solution $\mathbf{s}$ of length i is always the extension of a partial solution $\mathbf{s}'$ of length $i-1$.
In general, the cost for extending a partial solution $(\mathbf{s}', i-1)$ into another $(\mathbf{s}, i)$ depends on some information of them. By using a notation similar to that introduced in Section 4.3.2, we say that the state $[\mathbf{s}, i]$ of the partial solution $(\mathbf{s}, i)$ consists of all information needed for its expansion. As in general the state $[\mathbf{s}, i]$ contains less information than $(\mathbf{s}, i)$, it represents an equivalence class, eventually independent from i. Hence, the score for extending $(\mathbf{s}', i-1)$ into $(\mathbf{s}, i)$ is a function $S([\mathbf{s}'], [\mathbf{s}])$ of their states. For this reason, it is useful to define the set $Pred([\mathbf{s}])$ of all predecessors states of $[\mathbf{s}]$. $[\mathbf{s}']$ is a *predecessor* state of $[\mathbf{s}]$ if $\mathbf{s}'$, or an equivalent partial solution, is extended into $\mathbf{s}$, or an equivalent partial solution.
It is worth noticing that all partial solutions sharing the same state are recombined during the search algorithm because they are undistinguishable for the sake of expansion.
By exploiting the concept of state of a solution, the search criterion 5.15 can be restated as follows:

$$\begin{aligned}
\tilde{\mathbf{e}}^* &\approx \arg\max_{l} \max_{\mathbf{s}} \sum_{i=0}^{l} R(\mathbf{s}; \lambda, i) && (5.16)\\
&= \arg\max_{l} \max_{[\mathbf{s}]} \max_{\mathbf{s}' \in [\mathbf{s}]} \sum_{i=0}^{l} R(\mathbf{s}'; \lambda, i) && (5.17)\\
&= \arg\max_{l} \max_{[\mathbf{s}]} T([\mathbf{s}]; \lambda, l) && (5.18)
\end{aligned}$$

This formulation of the search criterion introduces the quantity $T([\mathbf{s}]; \lambda, i)$ representing the score of the best solution of length i and state $[\mathbf{s}]$.
This quantity can be recursively computed in terms of the so-called *extension* score $S([\mathbf{s}'], [\mathbf{s}])$, as follows:

base: $i = 0$

$$T([\mathbf{s}];\lambda,0) = \max_{\mathbf{s}'\in[\mathbf{s}]} R(\mathbf{s}';\lambda,0) \tag{5.19}$$

step: $i>0$

$$T([\mathbf{s}];\lambda,i) = \max_{[\mathbf{s}']\in Pred([\mathbf{s}])} T([\mathbf{s}'];\lambda,i-1)+S([\mathbf{s}'],[\mathbf{s}]) \tag{5.20}$$

5.1.3 Discussion on log-linear models

Real parameters λ weighting feature functions generalize those applied in the search criterion (4.5) to balance the contribution of the language and translation models.

Other great advantages of log-linear models are their mathematically soundness and the possibility to use any kind of features, regarded as important for the sake of translation.

Flexibility is also large because features are not required to be probability measures, overcoming the requirement of the source-channel based SMT model to have an exact factorization of the distributions.

If feature functions cannot be decomposed as suggested in Section 5.1.2, and, hence, cannot be embedded in the decoder, they can be added to the model after the decoding through a rescoring procedure. This technique consist of recomputing the global score of a set of alternative hypotheses, stored in a N-best list or in a WG, and extracting the new best translation.

Finally, it is worth remarking that any SMT model based on the source-channel framework can always be interpreted as a log-linear model.

The main issue arising from the use of log-linear models is the choice of the feature functions $h_1^M(\tilde{\mathbf{e}},\mathbf{f},\mathbf{a})$. Identifying best feature functions, which is an hard problem in many classification tasks of NLP, is also hard in SMT because feature functions should have the given form described in Section 5.1.2 to permit an efficient search.

To overcome this restriction it is possible to perform translation in two steps.

First simple log-linear model is defined with decomposable feature functions and a set of translation hypotheses are searched by means of criterion (5.9). Then, a more complex model with other features is used to re-rank the hypotheses. In this way, additional features are no longer required to be decomposable. A second issue related to the log-linear models is the estimation of the feature weights λ, which will be tackled in Section 5.3.

5.2 The source-channel based SMT model as log-linear model

In this Section we show that the SMT model presented in the previous Chapter can be seen as a log-linear model. First of all, we give a different description for the generative process of the translation hypotheses. Then we describe the model itself as a log-linear model. Finally, we prove that the generic search algorithm of Section 5.1.2 applied to this model is equivalent to the search algorithm presented in Section 4.3.2.

5.2.1 Generative process

Given $\mathbf{f} = f_1, \ldots, f_m$, an input string of m words in the source language, the generative process of a translation $\tilde{e}_1, \ldots, \tilde{e}_l$ consists of the following steps:

i. an index i is set to 0

ii. a fertility ϕ_0 is chosen

iii. a set π_0 of ϕ_0 positions of the source string is selected

iv. i is incremented by 1

v. a target phrase $\tilde{e}_i$ is selected within a dictionary $\tilde{\mathcal{E}}$

vi. a fertility ϕ_i is chosen

vii. a set π_i of ϕ_i *uncovered*[1] positions of the source string are selected

viii. steps (iv.)-(viii.) are repeated until all input positions are covered.

An "empty" target phrase $\tilde{e}_0$ corresponding to a virtual ε is introduced for the sake of simplifying the notation. If l is the number of repetitions of steps (iv.)-(viii.), the whole process results in a sequence of target phrases $\tilde{\mathbf{e}} = \tilde{e}_0^l$, fertilities $\phi = \phi_0^l$, and sets of positions $\pi = \pi_0^l$. As in step (viii.) only uncovered positions can be selected, permutations π_0^l induce a partition of $\mathbf{f}$. Moreover, there is an unambiguous correspondence between positions and words in $\mathbf{f}$; $f(\pi_i)$ will denote the set of ϕ_i words identified by π_i, and $f(\pi_{i,h})$ the corresponding h-th word.

Words $f(\pi_i), i > 0$ are those translated by the target phrase $\tilde{e}_i$; words $f(\pi_0)$ are not translated by any target phrases. If $\phi_0 = 0$, no input word remains untranslated.

If we set the constraint of only choosing consecutive positions in step (vii.), each set $f(\pi_i), i > 0$ of ϕ_i source words can be seen as a phrase.

In Section 4.2.4, where the generative process of the source-channel approach has been described, the set $f(\pi_i)$ has been named tablet and denoted with τ_i. There, words f_j are chosen within a dictionary (see Section 4.2.4), and, hence, they are variables. In the direct approach they are instead fixed. To highlight this difference, we replace notation of tablet τ_i with $f(\pi_i)$, although they are strictly related.

Moreover, any total or partial alignment $\mathbf{a} = (\phi, \pi)$ between $\mathbf{f}$ and $\tilde{\mathbf{e}}$ can be again denoted with a pair (ϕ, π).

It is worth remarking that an output sequence $\tilde{\mathbf{e}}$ can be generated from $\mathbf{f}$ in many different ways, each one corresponding to a specific (partial) alignment $\mathbf{a}$. On the other hand, a choice $a_0^i = (\phi_0^i, \pi_0^i)$ corresponds to a (partial) *compatible*

[1] A position is not covered, or uncovered, if it is not yet selected in any previous set $\pi_j, j < i$.

alignment between $\mathbf{f}$ and $\tilde{e}_0^i$ if π_0^i is a partition of the source positions $\bigcup_{t=0}^{i} \pi_t \subseteq \{1,\ldots,m\}$.

5.2.2 The SMT source-channel model as log-linear model

The phrase-based model has been described in Chapter 4 under the source-channel approach, showing that the maximization function (**??**) $Q(\mathbf{s}) = Q(\tilde{\mathbf{e}},\mathbf{f},\mathbf{a}) = Q(\tilde{\mathbf{e}},\tau,\phi,\pi)$ in the search criterion consists of the product of the language model $\Pr(\tilde{\mathbf{e}})$ and the translation model $\Pr(\tau,\phi,\pi \mid \tilde{\mathbf{e}})$, which is further decomposed in terms of fertility model, lexicon model, and distortion model. Moreover, we introduced suitable parameter for the sake of optimization of the whole model. The logarithm of $Q(\mathbf{s})$ is the following:

$$\begin{aligned} \log Q(\mathbf{s}) &= \log Q(\tilde{\mathbf{e}},\tau,\phi,\pi) && (5.21)\\ &= \lambda_{lm}\log\Pr(\tilde{\mathbf{e}}) \\ &\quad +\lambda_{fert}\log\Pr(\phi \mid \tilde{\mathbf{e}}) \\ &\quad +\lambda_{lex}\log\Pr(\tau \mid \phi,\pi,\tilde{\mathbf{e}}) \\ &\quad +\lambda_{dist}\Pr(\pi \mid \phi) && (5.22) \end{aligned}$$

By associating feature functions with the logarithm of each of the submodels, and corresponding feature weights as follows:

$$\begin{aligned} h_1(\mathbf{s}) &= \log\Pr(\tilde{\mathbf{e}}) \qquad \lambda_1 = \lambda_{lm} && (5.23)\\ h_2(\mathbf{s}) &= \log\Pr(\phi \mid \tilde{\mathbf{e}}) \qquad \lambda_2 = \lambda_{lex} && (5.24)\\ h_3(\mathbf{s}) &= \log\Pr(\tau \mid \phi,\pi,\tilde{\mathbf{e}}) \qquad \lambda_3 = \lambda_{fert} && (5.25)\\ h_4(\mathbf{s}) &= \log\Pr(\pi \mid \phi) \qquad \lambda_4 = \lambda_{dist} && (5.26) \end{aligned}$$

the maximization we obtain the same form 5.5 of the direct approach:

$$R(\mathbf{s};\lambda) = \log Q(\mathbf{s}) = \sum_{r=1}^{4} \lambda_r h_r(\mathbf{s}) \qquad (5.27)$$

5.2.3 The search algorithm of the SMT source-channel model

The search algorithm for the MT model presented in Section 4.3 can be described from the point of view of the direct approach. In order to prove that, we show that feature functions $h_1^M(\mathbf{s})$ of the MT model satisfy Equations 5.10 and 5.12.

$$\begin{aligned}
h_1(\mathbf{s}) = h_1(\tilde{\mathbf{e}},\mathbf{f},\phi,\pi) &= \log \Pr(\tilde{\mathbf{e}} = \tilde{e}_1,\ldots,\tilde{e}_l) \\
&= \log \prod_{i=1}^{l} p(k_i)\, p(\tilde{e}_i \mid \tilde{e}_{i-2},\tilde{e}_{i-1}) \\
&= \sum_{i=1}^{l} \log p(k_i) + \log p(\tilde{e}_i \mid \tilde{e}_{i-2},\tilde{e}_{i-1}) \quad (5.28) \\
h_1(\mathbf{s};0) = h_1(\tilde{\mathbf{e}},\mathbf{f},\phi,\pi;0) &= 0 \quad (5.29) \\
h_1(\mathbf{s};i) = h_1(\tilde{\mathbf{e}},\mathbf{f},\phi,\pi;i) &= \log p(k_i) + \\
&\quad \log p(\tilde{e}_i \mid \tilde{e}_{i-2},\tilde{e}_{i-1}) \quad \forall\ 1 \le i \le l \quad (5.30)
\end{aligned}$$

$$\begin{aligned}
h_2(\mathbf{s}) = h_2(\tilde{\mathbf{e}},\mathbf{f},\phi,\pi) &= \log \Pr(\phi \mid \tilde{\mathbf{e}}) \\
&= \log p(\phi_0 \mid m - \phi_0) \prod_{i=1}^{l} p(\phi_i \mid \tilde{e}_i) \\
&= \log p(\phi_0 \mid m - \phi_0) + \\
&\quad \sum_{i=1}^{l} \log \Pr(\phi_i \mid \tilde{e}_i) \quad (5.31) \\
h_2(\mathbf{s};0) = h_2(\tilde{\mathbf{e}},\mathbf{f},\phi,\pi;0) &= \log p(\phi_0 \mid m - \phi_0) \quad (5.32) \\
h_2(\mathbf{s};i) = h_2(\tilde{\mathbf{e}},\mathbf{f},\phi,\pi;i) &= \log p(\phi_i \mid \tilde{e}_i) \qquad \forall\ 1 \le i \le l \quad (5.33)
\end{aligned}$$

$$\begin{aligned}
h_3(\mathbf{s}) = h_3(\tilde{\mathbf{e}},\mathbf{f},\phi,\pi) &= \log \Pr(\mathbf{f} \mid \phi,\pi,\tilde{\mathbf{e}}) \\
&= \log p(f(\pi_0) \mid \phi_0) \prod_{i=1}^{l} p(f(\pi_i) \mid \phi_i,\tilde{e}_i)
\end{aligned}$$

$$\begin{aligned}
&= \log p(f(\pi_0) \mid \phi_0) + \\
&\quad \sum_{i=1}^{l} \log p(f(\pi_i) \mid \phi_i, \tilde{e}_i) && (5.34)\\
h_3(\mathbf{s};0) = h_3(\tilde{\mathbf{e}},\mathbf{f},\phi,\pi;0) &= \log p(f(\pi_0) \mid \phi_0, \tilde{e}_0)\\
&= \log \prod_{h=1}^{\phi_0} p(f(\pi_{0,h}) \mid \tilde{e}_0)\\
&= \sum_{h=1}^{\phi_0} \log p(f(\pi_{0,h}) \mid \tilde{e}_0) && (5.35)\\
h_3(\mathbf{s};i) = h_3(\tilde{\mathbf{e}},\mathbf{f},\phi,\pi;i) &= \log p(f(\pi_i) \mid \phi_i, \tilde{e}_i) \qquad \forall\ 1 \le i \le l && (5.36)
\end{aligned}$$

$$\begin{aligned}
h_4(\mathbf{s}) = h_4(\tilde{\mathbf{e}},\mathbf{f},\phi,\pi) &= \log \Pr(\pi \mid \phi)\\
&= \log \Pr(\pi_0 \mid \phi_0) \prod_{i=1}^{l} \Pr(\pi_i \mid \phi_i, \pi_0^{i-1})\\
&= \log p\left(\frac{1}{\phi_0!}\right) \prod_{i=1}^{l} p(\pi_i \mid \phi_i, \bar{\pi}_i)\\
&= \log p\left(\frac{1}{\phi_0!}\right) + \sum_{i=1}^{l} \log p(\pi_i \mid \phi_i, \bar{\pi}_i) && (5.37)\\
h_4(\mathbf{s};0) = h_4(\tilde{\mathbf{e}},\mathbf{f},\phi,\pi;0) &= \log p\left(\frac{1}{\phi_0!}\right) = -\log p(\phi_0!) && (5.38)\\
h_4(\mathbf{s};i) = h_4(\tilde{\mathbf{e}},\mathbf{f},\phi,\pi;i) &= \log p(\pi_i \mid \phi_i, \bar{\pi}_i) \qquad \forall\ 1 \le i \le l && (5.39)
\end{aligned}$$

It is worth noticing that $h_3(\mathbf{s};i) = h_4(\mathbf{s};i) = 0$ if $\phi_i = 0$.

The previous definition of the features functions h_1^4 induces the decomposition of $R(\mathbf{s};\lambda)$ in the form of equations (5.11) and (5.13), and, hence, the search criterion (5.15) is valid and the corresponding algorithm can be applied.

By looking at equations (5.28-5.39), we can observe that the state of the partial solution $(\mathbf{s},i) = (\tilde{\mathbf{e}},\mathbf{f},\phi,\pi;i)$ is $[\mathbf{s},i] = [\tilde{\mathbf{e}},\mathbf{f},\phi,\pi;i] = (C,\bar{\pi}_i,\tilde{e}_i,\tilde{e}_{i-1})$, where $C = \bigcup_{t=0,\ldots,i} \pi_t$. In fact, these are the only information needed to compute the score of an extension of $(\mathbf{s},i)$.

The first step of the generative process produces only partial solutions $(\mathbf{s},0)$ of length 0 with $[\mathbf{s},0]=(\pi_0,\bar{\pi}_0,\varepsilon,\varepsilon)$. Hence, only one partial solution of length 0 exists sharing a given state $[\mathbf{s}]=(\pi,\bar{\pi},\varepsilon,\varepsilon)$. This means that:

$$\begin{aligned} T([\mathbf{s}];\lambda,0) &= T([\pi,\bar{\pi},\varepsilon,\varepsilon;\lambda,0]) \\ &= R(\tilde{\mathbf{e}},\mathbf{f},\phi,\pi;0) \\ &= \sum_{r=1}^{4}\lambda_r h_r(\tilde{\mathbf{e}},\mathbf{f},\phi,\pi;0) \\ &= \lambda_2 \log p(\phi_0 \mid m-\phi_0) \\ &\quad + \lambda_3 \log p(f(\pi_0) \mid \phi_0,\tilde{e}_0) \\ &\quad - \lambda_4 \log p(\phi_0!) \end{aligned} \tag{5.40}$$

In the generic i-th step of the generative process the set $Pred([\mathbf{s}])$ of a partial solution $(\mathbf{s},i)=(\tilde{\mathbf{e}},\mathbf{f},\phi,\pi;i)$ of state $[\mathbf{s}]=(\mathcal{C},\bar{\pi},\tilde{e}_i,\tilde{e}_{i-1})$ contains states $[\mathbf{s}']=(\mathcal{C}\setminus\pi_i,\bar{\pi}',\tilde{e}_{i-1},\tilde{e}'')$. The corresponding expansion score is:

$$\begin{aligned} S([\mathbf{s}'],[\mathbf{s}]) &= S(\mathcal{C},\pi_i,\bar{\pi}_i,\tilde{e}_i,\tilde{e}_{i-1},\tilde{e}_{i-2}) \\ &= \sum_{r=1}^{4}\lambda_r h_r(\tilde{\mathbf{e}},\mathbf{f},\phi,\pi;i) \\ &= \lambda_1(\log p(k_i)+\log p(\tilde{e}_i \mid \tilde{e}_{i-2},\tilde{e}_{i-1})) \\ &\quad + \lambda_2 \log p(\phi_i \mid \tilde{e}_i) \\ &\quad + \lambda_3 \log p(f(\pi_i) \mid \phi_i,\tilde{e}_i) \\ &\quad + \lambda_4 \log p(\pi_i \mid \phi_i,\bar{\pi}_i) \end{aligned} \tag{5.41}$$

Notice that ϕ_i is univocally determined by π_i.

As the maximization in the equation (5.20) becomes a maximization over generic $\tilde{e}''$, $\emptyset\subseteq\pi_i\subseteq\mathcal{C}$, and $\bar{\pi}'$ so that $\bar{\pi}_i=\bar{\pi}$, we rewrite:

$$\begin{aligned} T([\mathbf{s}];\lambda,i) &= T((\mathcal{C},\bar{\pi},\tilde{e},\tilde{e}');\lambda,i) \\ &= \max_{\tilde{e}'',\emptyset\subseteq\pi_i\subseteq\mathcal{C}:\bar{\pi}_i=\bar{\pi},\bar{\pi}'} T((\mathcal{C}\setminus\pi_i,\bar{\pi}',\tilde{e}',\tilde{e}'');\lambda,i-1) \\ &\qquad\qquad S(\mathcal{C},\pi_i,\bar{\pi}_i,\tilde{e},\tilde{e}',\tilde{e}'') \end{aligned}$$

$$= \max_{\tilde{e}'', \emptyset \subseteq \pi_i \subseteq C : \bar{\pi}_i = \bar{\pi}, \bar{\pi}'} T\big((C \setminus \pi_i, \bar{\pi}', \tilde{e}', \tilde{e}''); \lambda, i-1\big) \cdot R(\tilde{\mathbf{e}}, \mathbf{f}, \phi, \pi; \lambda, i) \quad (5.42)$$

This recursive formulation of the search criterion consisting of equation (5.40) and (5.42) perfectly corresponds to that described in Section 4.3, a part from the application of the logarithmic function.

5.3 Optimization of the feature weights

Optimal estimate of the weights can be achieved following two strategies. The former is the maximum entropy solution corresponding to values λ, which maximizes the log-likelihood over a training sample:

$$\lambda^* = \arg\max_{\lambda} \sum_{(\tilde{\mathbf{e}}, \mathbf{f}, \mathbf{a})} \log p_\lambda(\tilde{\mathbf{e}}, \mathbf{a} \mid \mathbf{f}) \quad (5.43)$$

Unfortunately, a closed-form solution of this criterion does not exist. An iterative procedure converging to the solutions have been proposed by [19] and improved by [20].
In place of the criterion (5.43), [60] recently proposed to estimate parameters by directly minimizing the number of translation errors.

5.3.1 Minimum Error Training

We assume that a function $E_D(\lambda)$ is available, which measures the translation errors made by running a model defined by parameter values λ on a development set D. Hence, parameters are searched by:

$$\lambda_* = \arg\min_{\lambda} E_D(\lambda) \quad (5.44)$$

Unlike the log-likelihood criterion (5.43), the objective function $E_D(\cdot)$ might have many local minima. Neverthelss, an approximate algorithm, the *simplex method* [66], is used which requires relatively few function evaluations.

If the whole translation process is time-costly, it is possible to translate once only, to extract the set of N-best hypotheses, and to compute $E_D(\lambda)$ over them after rescoring.

Chapter 6

Spoken Language Translation

This Chapter reports recent research activity in the field of Spoken Language Translation. Two SLT systems have been developed, which extend in different ways the MT system presented in Chapter 4.

The first system exploits a list of N-best transcriptions provided by the ASR system. Two additional ASR features, namely the acoustic and language models, are used for re-score the best translation of each input transcription. The combination of speech recognition and translation models results in a statistical log-linear model of eight feature functions.

A tighter integration between speech recognition and translation is achieved by defining an other statistical log-linear model working on a particular word graph, called *confusion network*, generated by the ASR system. The search algorithm for the MT system is properly extended to work with this kind of input.

The main advantage we expected from the last approach is to save computation time, without loosing translation quality. In fact, for each speech utterance, the system based on confusion network performs just one decoding pass, while the system based on N-best transcription hypotheses need to translate N times.

This Chapter is organized as follows. The next Section gives a brief overview of the main approaches to SLT. Section 6.2 introduces the Automatic Speech

Recognition (ASR) task. In Sections 6.4, 6.5 and 6.6 two SLT systems developed at ITC-irst are presented. In Section 6.8 the comparison of the two systems is provided on an Italian-English speech translation task in a tourism domain.

6.1 Previous work

First attempts to tackle the SLT task simply performed speech recognition and translation sequentially: first the ASR module recognized the spoken text and then the MT module translated it [83, 75]. In this way, however, recognition errors cannot be handled by the translation errors.

Experience in speech recognition tells that the use of N-best transcriptions reduce the word error rate. Hence, following approaches tried to exploit a set of transcription alternatives as input for the MT systems. Useful supplementary information available from speech recognition, like the acoustic and language models, can improve translation performance if employed properly. For instance, [92] presented a log-linear model, which combines features from ASR and MT. Independently, we proposed similar model, which instead exploits the phrase-based translation model of Chapter 4.

A tighter integration of speech recognition and translation was suggested in [51], which needs some local approximation and works for monotone alignments only. Along this way is the work of [71] which presents improvement in translation quality by using a word lattice as interface between ASR and MT. Similarly, but independently, we present a second model which exploits an approximation of the word graph generated by the ASR system.

Stochastic Finite-State Transducers have been also applied in SLT [17]. Transducers can be easily integrated with the conventional ASR models and allow the use of simple traditional Viterbi beam-search techniques translate speech signal. However, this approach has the great disadvantage that it can be applied

only in very limited domains.

6.2 Automatic Speech Recognition

According to [36], at a very basic form, a speech recognizer is a device that automatically transcribes speech into text. The recognized words can be the final results, as for applications such as commands and control, data entry, and document preparation. They can also serve as the input to further linguistic processing in order to achieve speech understanding, spoken information retrieval, speech translation etc.

Without loss of generality we may assume that the speech signal is represented by a sequence of symbols $\mathbf{o} = o_1, o_2, ..., o_T$ taken from some alphabet O, where the index corresponds to the time in which the symbol has been generated.

The recognizer should search for a word string $\mathbf{f}^*$ satisfying:

$$\mathbf{f}^* = \arg\max_{\mathbf{f}} \Pr(\mathbf{f}|\mathbf{o}) \tag{6.1}$$

where $\Pr(\mathbf{f}|\mathbf{o})$ is the probabilities that the word sequence $\mathbf{f}$ were spoken, given that the evidence $\mathbf{o}$ was observed, and $\mathbf{f}$ denotes any string of a known vocabulary $\mathcal{F}$.

The well-known Bayes formula of probability theory allows to rewrite the right-hand side probability of (6.1) as:

$$\Pr(\mathbf{f}|\mathbf{o}) = \frac{\Pr(\mathbf{f}) \cdot \Pr(\mathbf{o}|\mathbf{f})}{\Pr(\mathbf{o})} \tag{6.2}$$

where the *language model* $\Pr(\mathbf{f})$ is the probability that the word sequence $\mathbf{f}$ will be uttered, the *acoustic model* $\Pr(\mathbf{o}|\mathbf{f})$ is the probability of observing the acoustic evidence $\mathbf{o}$ when the speaker says $\mathbf{f}$, and $\Pr(\mathbf{o}) = \sum_{\mathbf{f}'} \Pr(\mathbf{f}') \cdot \Pr(\mathbf{o}|\mathbf{f}')$ is the average probability that $\mathbf{o}$ will be observed.

Since the variable $\mathbf{o}$ is fixed in the maximization in (6.1) the denominator can

be discarded and (6.1) becomes:

$$\mathbf{f}^* = \arg\max_{\mathbf{f}} \Pr(\mathbf{f}) \cdot \Pr(\mathbf{o}|\mathbf{f}) \tag{6.3}$$

As $\Pr(\mathbf{o}|\mathbf{f})$ is usually the product of many more factors than $\Pr(\mathbf{f})$, the decision for a word sequence would be dominated by the acoustic scores and the language model would have hardly influence. To balance this very large difference in the probability values, it is usual to use two exponential weights λ_{ac} and λ_{lm} for the acoustic model and for the language model, respectively. Thus (6.3) can be written as follows:

$$\mathbf{f}^* = \arg\max_{\mathbf{f}} \Pr(\mathbf{f})^{\lambda_{lm}} \cdot \Pr(\mathbf{o}|\mathbf{f})^{\lambda_{ac}} \tag{6.4}$$

From the point of view of the Maximum Entropy framework (see Chapter 5), the distribution $\Pr(\mathbf{f}|\mathbf{o})$ can be easily redefined as a log-linear model as follows:

$$h_{lm}(\mathbf{f},\mathbf{o}) = \log \Pr(\mathbf{f}) \tag{6.5}$$

$$h_{ac}(\mathbf{f},\mathbf{o}) = \log \Pr(\mathbf{o} \mid \mathbf{f}) \tag{6.6}$$

$$\Pr(\mathbf{o} \mid \mathbf{f}) = \frac{\exp(\lambda_{lm} h_{lm}(\mathbf{f},\mathbf{o}) + \lambda_{ac} h_{ac}(\mathbf{f},\mathbf{o}))}{\sum_{\mathbf{f}'} \exp(\lambda_{lm} h_{lm}(\mathbf{f}',\mathbf{o}) + \lambda_{ac} h_{ac}(\mathbf{f}',\mathbf{o}))} \tag{6.7}$$

and, hence, the maximization formula becomes

$$\mathbf{f}^* = \arg\max_{\mathbf{f}} (\lambda_{lm} h_{lm}(\mathbf{f},\mathbf{o}) + \lambda_{ac} h_{ac}(\mathbf{f},\mathbf{o})) \tag{6.8}$$

6.2.1 ITC-irst ASR system

The ASR system employed for experiments has been developed at ITC-irst since 1990s [15].

The speech transcription engine is the core module of the system. It is is based on a single-step time-synchronous Viterbi decoder [14, 15], where the LM is mapped into a static network with a shared-tail topology [1]. A *beam search* technique [16] avoids an exhaustive search of a huge space by pruning less promising hypotheses on the basic of a local estimation.

The acoustic model is based on context-dependent triphones, speaker-independent, continuous-density Hidden Markov Models (HMM) [68]. Acoustic training was performed with Baum-Welch reestimation. The language model is based on trigram statistics smoothed by combining non-linear discounting and interpolation with lower-order models [23].

Output formats

As any other ASR system, the ITC-irst system outputs the *1-best transcription*, which maximizes the probability score $\Pr(\mathbf{f}|\mathbf{o})$. Moreover, all transcription hypotheses generated and not pruned during the decoding are stored in a *word-graph*.
As mentioned in Section 1.3, an SLT system could benefit from the availability of more transcription alternatives. Unfortunately, the word-graph produced can be very large and not weasy to manage. Hence, further modules have been developed at ITC-irst, which approximate the word graphs in two ways:

- a list of the *N-best transcriptions* [81];
- a *confusion network* (CN) [46]; in particular, a CN is a word-graph itself but it is much more compact and contains more paths than the word-graph generated by the ASR system.

6.3 Spoken Language Translation

From a statistical point of view, SLT can be considered as an extension of SMT. In particular we are interested in finding the best translation of an utterance rather than of a text. Again, we use a probability measure to express the closeness of possible output strings to the input utterance. Formally, the SLT problem can be stated as follows: given the acoustic observation $\mathbf{o}$ in the source

language, find the string $\tilde{\mathbf{e}}^*$ in the target language, which maximizes the probability $\Pr(\tilde{\mathbf{e}} \mid \mathbf{o})$

$$\tilde{\mathbf{e}}^* = \arg\max_{\tilde{\mathbf{e}}} \Pr(\tilde{\mathbf{e}} \mid \mathbf{o}) \tag{6.9}$$

If $\mathbf{o}$ is the vector representing the acoustic observation of the input utterance, we define $\mathcal{F}(\mathbf{o})$ as the set of all transcription hypotheses provided by the available ASR decoder. In the case of the ITC-irst ASR system, $\mathcal{F}(\mathbf{o})$ consists of a word graph, as stated in the previous Section. The SLT search criterion can be rewritten as follows:

$$\tilde{\mathbf{e}}^* = \arg\max_{\tilde{\mathbf{e}}} \sum_{\mathbf{f} \in \mathcal{F}(\mathbf{o})} \Pr(\tilde{\mathbf{e}}, \mathbf{f} \mid \mathbf{o}) \tag{6.10}$$

where $\mathbf{f}$ is an hidden variable representing any speech transcription hypothesis. This gives us the freedom of generating the best speech translation by considering the contribution of all transcription hypotheses. By comparing the search criterion in (6.10) for SLT with that of SMT (4.1), we notice a further level of complexity, consisting in the summation over $\mathcal{F}(\mathbf{o})$. A complete search over all transcription hypotheses $\mathbf{f}$ in $\mathcal{F}(\mathbf{o})$ is often hard to realize because:

- the size of the set $\mathcal{F}(\mathbf{o})$ is usually huge;
- the word graph produced by the ASR system is not easy to manage for the sake of extracting alternative speech transcriptions.

In order to efficiently overcome these problems, two strategies are followed. The first approach, discussed in Section 6.4, consists of approximating the search algorithm by reducing the set of transcription hypotheses considered. The second approach exploits a *confusion network*, described in Section 6.5, as an approximation of the word-graph.

6.4 N-best approach

To cope with the huge size of $\mathcal{F}(\mathbf{o})$, only a subset of all speech transcriptions is considered in the search criterion. A reasonable choice is to take the set of N most probable hypotheses with respect to the ASR model $\Pr(\mathbf{f} \mid \mathbf{o})$. In this case, $\mathcal{F}(\mathbf{o})$ is substituted by a subset $\mathcal{F}_N(\mathbf{o}) = \{\mathbf{f}_1, \ldots, \mathbf{f}_N\}$. By taking a maximum approximation over $\mathcal{F}_N(\mathbf{o})$, we get the search criterion:

$$\begin{aligned}
\tilde{\mathbf{e}}^* &\approx \arg\max_{\tilde{\mathbf{e}}} \sum_{n=1}^{N} \Pr(\tilde{\mathbf{e}}, \mathbf{f}_n \mid \mathbf{o}) && (6.11)\\
&\approx \arg\max_{\tilde{\mathbf{e}}} \max_{n=1,..,N} \Pr(\tilde{\mathbf{e}}, \mathbf{f}_n \mid \mathbf{o}) && (6.12)\\
&= \arg \max_{n=1,..,N} \max_{\tilde{\mathbf{e}}} \Pr(\tilde{\mathbf{e}}, \mathbf{f}_n \mid \mathbf{o}) && (6.13)
\end{aligned}$$

Under the reasonable assumption that $\tilde{\mathbf{e}}$ is stochastically independent from the acoustic observation $\mathbf{o}$, we can decompose $\Pr(\tilde{\mathbf{e}}, \mathbf{f} \mid \mathbf{o})$ in the following way:

$$\Pr(\tilde{\mathbf{e}}, \mathbf{f} \mid \mathbf{o}) = \Pr(\mathbf{f} \mid \mathbf{o}) \Pr(\tilde{\mathbf{e}} \mid \mathbf{f}) \tag{6.14}$$

and compute $\tilde{\mathbf{e}}^*$ as follows:

$$\begin{aligned}
\tilde{\mathbf{e}}^* &\approx \arg \max_{n=1,..,N} \max_{\tilde{\mathbf{e}}} \Pr(\mathbf{f}_n \mid \mathbf{o}) \Pr(\tilde{\mathbf{e}} \mid \mathbf{f}_n) && (6.15)\\
&= \arg \max_{n=1,..,N} \Pr(\mathbf{f}_n \mid \mathbf{o}) \max_{\tilde{\mathbf{e}}} \Pr(\tilde{\mathbf{e}} \mid \mathbf{f}_n) && (6.16)
\end{aligned}$$

In the above equation we can point out N problems of text translation (rightmost maximization), and one recombination problem of N results (leftmost maximization). Hence, the approximate search criterion (6.13) can be restated as follows:

$$\begin{aligned}
\tilde{\mathbf{e}}_n^* &= \arg\max_{\tilde{\mathbf{e}}} \Pr(\tilde{\mathbf{e}} \mid \mathbf{f}_n) \qquad n = 1, \ldots, N && (6.17)\\
\tilde{\mathbf{e}}^* &\approx \arg \max_{n=1,..,N} \Pr(\mathbf{f}_n \mid \mathbf{o}) \Pr(\tilde{\mathbf{e}}_n^* \mid \mathbf{f}_n) && (6.18)
\end{aligned}$$

In plain words: first the best translation $\tilde{\mathbf{e}}_n^*$ of each transcription hypothesis $\mathbf{f}_n$ is searched, then, the best translation $\tilde{\mathbf{e}}^*$ is selected among $\{\tilde{\mathbf{e}}_1^*, \ldots, \tilde{\mathbf{e}}_N^*\}$ according to its score weighte by the recognition probability $\Pr(\mathbf{f}_n \mid \mathbf{o})$.
This cascade algorithm works whatever ASR and MT systems are available, which provide scores for the best recognition and translation hypotheses, `ASR-score` and `MTscore`, respectively:

$$\tilde{\mathbf{e}}_n^* = \arg\max_{\tilde{\mathbf{e}}} \texttt{MTscore}(\tilde{\mathbf{e}}; \mathbf{f}_n) \qquad n = 1, \ldots, N \tag{6.19}$$

$$\tilde{\mathbf{e}}^* \approx \arg\max_{n=1,..,N} \texttt{ASRscore}(\mathbf{f}_n; \mathbf{o})\, \texttt{MTscore}(\tilde{\mathbf{e}}_n^*; \mathbf{f}_n) \tag{6.20}$$

6.4.1 The N-best based SLT log-linear model

The search criterion (6.13) can be detailed through the introduction of the hidden alignment variable $\mathbf{a}$, and by taking an additional maximum approximation:

$$\tilde{\mathbf{e}}^* \approx \arg\max_{n=1,..,N} \max_{\tilde{\mathbf{e}}} \sum_{\mathbf{a} \in \mathcal{A}(\mathbf{f}_n, \tilde{\mathbf{e}})} \Pr(\tilde{\mathbf{e}}, \mathbf{f}_n, \mathbf{a} \mid \mathbf{o}) \tag{6.21}$$

$$\approx \arg\max_{n=1,..,N} \max_{\tilde{\mathbf{e}}} \max_{\mathbf{a} \in \mathcal{A}(\mathbf{f}_n, \tilde{\mathbf{e}})} \Pr(\tilde{\mathbf{e}}, \mathbf{f}_n, \mathbf{a} \mid \mathbf{o}) \tag{6.22}$$

The resulting SLT model $\Pr(\tilde{\mathbf{e}}, \mathbf{f}, \mathbf{a} \mid \mathbf{o})$ can be expressed as a log-linear model of six features, $h_r(\tilde{\mathbf{e}}, \mathbf{f}, \mathbf{a}, \mathbf{o})$, $r = 1, \ldots, 6$. The first 4 features functions are trivial extensions of those defining the MT model $\Pr(\tilde{\mathbf{e}}, \mathbf{a} \mid \mathbf{f})$, introduced in Section 5.2:

$$h_r(\tilde{\mathbf{e}}, \mathbf{f}, \mathbf{a}, \mathbf{o}) = h_r(\tilde{\mathbf{e}}, \mathbf{f}, \mathbf{a}) \qquad r = 1, \ldots, 4 \tag{6.23}$$

as they do depend on the acoustic observation $\mathbf{o}$.
Two additional features functions are defined as the logarithm of the source language model $\Pr(\mathbf{f})$ and of the acoustic model $\Pr(\mathbf{o} \mid \mathbf{f})$ introduced in Section 6.2:

$$h_5(\tilde{\mathbf{e}}, \mathbf{f}, \mathbf{a}, \mathbf{o}) = h_5(\mathbf{f}, \mathbf{o}) = \log \Pr(\mathbf{f}) \tag{6.24}$$

$$h_6(\tilde{\mathbf{e}}, \mathbf{f}, \mathbf{a}, \mathbf{o}) = h_6(\mathbf{f}, \mathbf{o}) = \log \Pr(\mathbf{o} \mid \mathbf{f}) \tag{6.25}$$

which are independent from $\tilde{\mathbf{e}}$ and $\mathbf{a}$.
Hence, the SLT model $\Pr(\tilde{\mathbf{e}}, \mathbf{f}, \mathbf{a} \mid \mathbf{o})$ is expressed as follows:

$$\begin{aligned}
\Pr(\tilde{\mathbf{e}}, \mathbf{f}, \mathbf{a} \mid \mathbf{o}) &= \frac{\exp\left\{\sum_{r=1}^{6} \lambda_r h_r(\tilde{\mathbf{e}}, \mathbf{f}, \mathbf{a}, \mathbf{o})\right\}}{\sum_{\tilde{\mathbf{e}}', \mathbf{f}' \mathbf{a}'} \exp\left\{\sum_{r=1}^{6} \lambda_r h_r(\tilde{\mathbf{e}}, \mathbf{f}, \mathbf{a}, \mathbf{o})\right\}} && (6.26) \\
&\propto \exp\left\{\sum_{r=1}^{6} \lambda_r h_r(\tilde{\mathbf{e}}, \mathbf{f}, \mathbf{a}, \mathbf{o})\right\} && (6.27) \\
&= \exp\left\{\sum_{r=1}^{4} \lambda_r h_r(\tilde{\mathbf{e}}, \mathbf{f}, \mathbf{a})\right\} \exp\left\{\sum_{r=5}^{6} \lambda_r h_r(\mathbf{f}, \mathbf{o})\right\} && (6.28)
\end{aligned}$$

The last equation higlights the separate contributions given by the SMT model $\Pr(\tilde{\mathbf{e}}, \mathbf{a} \mid \mathbf{f})$ and the ASR model $\Pr(\mathbf{f} \mid \mathbf{o})$.
Hence, the search criterion (6.13) for the N-best approach can be rewritten as:

$$\begin{aligned}
(\tilde{\mathbf{e}}_n^*, \mathbf{a}_n^*) &= \arg\max_{\tilde{\mathbf{e}}} \max_{\mathbf{a} \in \mathcal{A}(\mathbf{f}_n, \tilde{\mathbf{e}})} \sum_{r=1}^{4} \lambda_r h_r(\tilde{\mathbf{e}}, \mathbf{a}, \mathbf{f}_n, \mathbf{o}) \qquad \forall\ 1 \leq n \leq N && (6.29) \\
\tilde{\mathbf{e}}^* &= \arg\max_{n=1,..,N} \left(\sum_{r=5}^{6} \lambda_r h_r(\mathbf{f}_n, \mathbf{o}) + \sum_{r=1}^{4} \lambda_r h_r(\tilde{\mathbf{e}}_n^*, \mathbf{a}_n^*, \mathbf{f}_n, \mathbf{o}) \right) && (6.30)
\end{aligned}$$

where $\tilde{\mathbf{e}}_n^*$ and $\mathbf{a}_n^*$ are the best translation and alignment for $\mathbf{f}_n$. Notice that the best score R_n^*, corresponding to $\tilde{\mathbf{e}}_n^*$, can be plugged directly into (6.30), causing $\mathbf{a}_n^*$ to be useless.
The definition of the N-best SLT model in terms of features of both the MT and ASR systems is useful because it allows the global optimization of parameters λ.

6.4.2 Analysys of the complexity

The algorithm solving the search criterion for the N-best decoder is reported in Figure 6.1. After extraction of the N-best transcription hypotheses from a WG (line 1), each hypothesis is translated and scored (line 5). At each step, the best translation (*best*) and its score (*bestsc*) are possibly updated (lines 6-8).

NBEST-SLT

1 $\mathcal{F}_N(\mathbf{o}) \leftarrow$ NBESTEXTRACTION$(\mathbf{o})$
2 $best \leftarrow 1$
3 $bestsc \leftarrow -\infty$
4 **for** $n = 1 \rightarrow N$
5 **do** sc = COMPUTEMTSCORE$(\mathbf{f}_n)$+ COMPUTEASRSCORE$(\mathbf{f}_n)$
6 **if** $sc > bestsc$
7 **then** $bestsc$=sc
8 $best$=n
9 GETSOLUTION($best$)

Figure 6.1: Search algorithm for the N-best based SLT system.

As the algorithm translates the N transcription hypotheses, its complexity is trivially $O(N\ g(m_{max}, |\mathcal{E}|))$, where m_{max} is the maximum string length within $\mathcal{F}_N$, and $g(m, |\mathcal{E}|)$ is the complexity of the algorithm for text translation (see Section 4.3.1). The complexity of NBESTEXTRACTION procedure is not taken into account; details about it can be found in [67] and papers referred in it.

6.5 Confusion Network

A Confusion Network (CN) is a word graph, called *sausage* [46], with the peculiarity that each path from the start to the end node has to go through all nodes. It can be represented as a set of words $\mathbf{w}$ and probabilities $\mathbf{p}$ placed into m columns of different depths.

The following notation $\mathcal{G}(\{1,\ldots,m\}, \mathbf{d}, \mathbf{w}, \mathbf{p})$ means that the CN has m columns and that the j-th column has depth d_j, for $j = 1,\ldots,m$. The shorter notation $\mathcal{G}(\{1,\ldots,m\}, \mathbf{w}, \mathbf{p})$ is used if the column depths are known. Moreover, $\mathcal{G}(\mathcal{C}, \mathbf{d}, \mathbf{w}, \mathbf{p})$, or $\mathcal{G}(\mathcal{C}, \mathbf{w}, \mathbf{p})$, consists of the sub-CN built up by taking only columns with indexes in $\mathcal{C} \subseteq \{1,\ldots,m\}$.

Each position (j,k) of the CN is associated with a word $w_{j,k}$ and a probability

era$_{0.997}$	cancello$_{0.995}$	ε$_{0.999}$	di$_{0.615}$	imbarco$_{0.999}$	...
è$_{0.002}$	vacanza$_{0.004}$	la$_{0.001}$	di$_{0.376}$	bar$_{0.001}$	
ε$_{0.001}$	ε$_{0.002}$		l'$_{0.002}$		
			...		
			ε$_{0.001}$		

Figure 6.2: Matrix representation of a confusion network generated from an Italian input utterance. Words and posterior probabilities are shown. The manual transcription of this utterance is "dove è il cancello d' imbarco".

value $p_{j,k}$, whose meaning is explained below. We always assume that words are inside the CN, i.e. $k \leq d_j$.

Figure 6.2 shows the matrix representation of an Italian input utterance extracted from the test set. The corresponding word graph is depicted in Figure A.2 in Appendix A.

A string $\mathbf{f} = f_1, \ldots, f_m$ is a realization of $\mathcal{G}(\{1,\ldots,m\}, \mathbf{d}, \mathbf{w}, \mathbf{p})$ if for all $j = 1,\ldots,m : f_j = w_{j,k_j}$ for some $k_j \in \{1,\ldots,d_j\}$. Viceversa, any choice $k_1,\ldots,k_m$ so that $k_j \in \{1,\ldots,d_j\}$ for $j = 1,\ldots,m$ defines the realization $\mathbf{f} = w_{1,k_1},\ldots,w_{m,k_m}$. In the graph representation of a CN a string $\mathbf{f}$ is a path from the start to the end node. In the following, $\mathcal{F}(\mathcal{G})$ will indicate the set of all realizations of $\mathcal{G}$.

A CN $\mathcal{G}(\{1,\ldots,m\}, \mathbf{d}, \mathbf{w}, \mathbf{p})$ is generated from a word graph produced by an ASR system starting from acoustic observation $\mathbf{o}$. Hence, the value $p_{j,k}$ associated with the word $w_{j,k}$ corresponds to the posterior probability $p_{j,k} = \Pr(w_{j,k} = f \mid \mathbf{o}, j)$ of having f at "position j given $\mathbf{o}$. It is worth noticing that $\Pr(f \mid \mathbf{o}, j)$ defines a probability measure over all words of the j-th column of the CN; formally, $\sum_{k=1}^{d_j} p_{j,k} = 1$ for $j = 1,\ldots,m$. A realization $\mathbf{f} = f_1,\ldots,f_m$ of $\mathcal{G}(\mathbf{w}, \mathbf{p}, \{1,\ldots,m\})$ is associated with the probability $\Pr(\mathbf{f} \mid \mathbf{o})$ of having $\mathbf{f}$ given $\mathbf{o}$, which can be factorized in terms of $\Pr(f \mid \mathbf{o}, j)$ as follows:

$$\Pr(\mathbf{f} \mid \mathbf{o}) = \prod_{j=1}^{m} \Pr(f_j \mid \mathbf{o}, j) \tag{6.31}$$

Notice that this decomposition assumes stochastic independence between the posterior probabilities of the single words.
The generation of the confusion network from the word graph can produce some special words, ε, which correspond to empty words. For the sake of simplicity, we assume that ε-words are completely undistiguishable from the other normal words, unless differently specified.

6.6 Confusion Network approach

Given a Confusion Network $\mathcal{G}$, the best translation $\tilde{\mathbf{e}}^*$ is searched through the criterion:

$$\tilde{\mathbf{e}}^* = \arg\max_{\tilde{\mathbf{e}}} \Pr(\tilde{\mathbf{e}} \mid \mathcal{G}) \tag{6.32}$$

$$= \arg\max_{\tilde{\mathbf{e}}} \sum_{\mathbf{a} \in \mathcal{A}(\mathcal{G}, \tilde{\mathbf{e}})} \Pr(\tilde{\mathbf{e}}, \mathbf{a} \mid \mathcal{G}) \tag{6.33}$$

where the search of maximum is performed over all compatible aligments $\mathcal{A}(\mathcal{G}, \tilde{\mathbf{e}})$ between $\mathcal{G}$ and $\tilde{\mathbf{e}}$. Later we will show that this criterion is equivalent to (6.10).
The substitution of the summation with the maximum operation introduces the usual approximation:

$$\tilde{\mathbf{e}}^* \approx \arg\max_{\tilde{\mathbf{e}}} \max_{\mathbf{a} \in \mathcal{A}(\mathcal{G}, \tilde{\mathbf{e}})} \Pr(\tilde{\mathbf{e}}, \mathbf{a} \mid \mathcal{G}) \tag{6.34}$$

The CN-based SLT model $\Pr(\tilde{\mathbf{e}}, \mathbf{a} \mid \mathcal{G})$, presented in subsection 6.6.3, is a straightforward extension of the the log-linear SMT model.

6.6.1 The generative process from a Confusion Network

The generative process of an output string $\tilde{\mathbf{e}}$ from an input string $\mathbf{f}$ described in Section 5.2.3 can be extended to exploit an input consisting of a CN.

As stated before, a word of a CN is identified with two indexes, one for the column and one for the position in the column. Hence, besides $\pi = \pi_0^l$, new sets $\psi = \psi_0^l$ are introduced, where $\psi_{i,h}$ represents the position of the word in the column $\pi_{i,h}$. As the constraints $\psi_{i,h} \in \{1,\ldots,d_{\pi_{i,h}}\}$ for all $i = 0,\ldots,l$ and for all $h = 0,\ldots,\phi_{i,h}$ have to be satisfied, ψ are dependent from π. The generative process described in Sectio 5.2.1 becomes:

i. an index i is set to 0

ii. a fertility ϕ_0 is chosen

iii. a set π_0 of ϕ_0 columns are selected

iv. a set ψ_0 of ϕ_0 non negative integers are chosen

v. i is incremented by 1

vi. a target phrase $\tilde{e}_i$ is selected within a dictionary $\tilde{\mathcal{E}}$

vii. a fertility ϕ_i is chosen

viii. a set π_i of ϕ_i *uncovered*[1] columns are selected

ix. a set ψ_i of ϕ_i non negative integers are chosen

x. steps (v.)-(x.) are repeated until all columns of the CN are covered.

This new generative process induces an augmented alignment $\mathbf{a} = a_0^l = (\phi, \pi, \psi)$ between the CN $\mathcal{G}(\{1,\ldots,m\}, \mathbf{d}, \mathbf{w}, \mathbf{p})$ and the output string $\tilde{\mathbf{e}}$. Each alignment identifies a specific realization $\mathbf{f} = f_1,\ldots,f_m$ of $\mathcal{G}$ so that for $j = 1,\ldots,m$ $f_j = w_{\pi_{i,h},\psi_{i,h}}\ \forall\ i,h$; in the following, we will denote $\mathbf{f} = \mathbf{w}(\mathbf{a}) = \mathbf{w}(\phi,\pi,\psi)$ the realization aligned with $\tilde{\mathbf{e}}$ through $\mathbf{a}$.

[1] A column is uncovered if it is not yet selected in any previous set π_j, $j < i$.

After i steps of the generative process, a partial alignment $a_0^i = (\phi_0^i, \pi_0^i, \psi_0^i)$ is induced between the subCN $\mathcal{G}(\mathcal{C}, \mathbf{w}, \mathbf{p})$, and $\tilde{e}_0^i$, where $\mathcal{C} = \bigcup_{t=0,\ldots,i} \pi_t$, or equivalently between the partial realization $\mathbf{w}(\mathbf{a}; i) = \mathbf{w}(\phi, \pi, \psi; i)$ and $\tilde{e}_0^i$. Moreover, $\mathbf{w}(\phi_i, \pi_i, \psi_i)$ will denote the set of words aligned with $\tilde{e}_i$, and $\mathbf{w}(\phi_i, \pi_{i,h}, \psi_{i,h}) = w_{\pi_{i,h},\psi_{i,h}}$ the corresponding h-th word. As done in the case of a single input string (see Section **??**) the triple $(\tilde{\mathbf{e}}, \mathcal{G}, \mathbf{a})$ will be often substituted with the fivetuple $(\tilde{\mathbf{e}}, \mathcal{G}, \phi, \pi, \psi)$, and, more generally, $(\tilde{\mathbf{e}}, \mathcal{G}, \mathbf{a}; i)$ with $(\tilde{\mathbf{e}}, \mathcal{G}, \phi, \pi, \psi; i)$. The shorthands $(\mathbf{s})$ and $(\mathbf{s}, i)$ will identify a complete solution of length l and a partial solution of length i, respectively.

$\mathcal{A}(\mathcal{G}, \tilde{\mathbf{e}})$ will denote the set of all compatible alignments between $\mathcal{G}$ and $\tilde{\mathbf{e}}$. It is trivial to prove that $\mathcal{A}(\mathcal{G}, \tilde{\mathbf{e}}) = \bigcup_{\mathbf{f} \in \mathcal{F}(\mathcal{G})} \mathcal{A}(\mathbf{f}, \tilde{\mathbf{e}})$, which states the equivalence between the search criterions (6.10) and (6.33).

6.6.2 Handling ε words

An important issue arises from the presence of ε-words in the CN $\mathcal{G}$ (see Section 6.5). Whereas they do not affect the generative process, they have to be handled very carefully in the definition of the feature functions, because they are not words actually.

Let us assume that $(\tilde{\mathbf{e}}, \mathcal{G}, \phi, \pi, \psi)$ is obtained during the generative process. The realization $\mathbf{w}(\phi, \pi, \psi)$ of length m can consist of less real words because it eventually might contain some ε-words. For this reason fertilities ϕ and permutations π and ψ might be modified. The related feature functions should take into account the difference between real and ε-words.

In order to give a correct definition of the SLT model, a further notation is introduced:

$$\hat{\phi}_i = \phi_i - \sum_{h=1}^{\phi_i} \delta(w_{\pi_{i,h},\psi_{i,h}} = \varepsilon) \tag{6.35}$$

$$\hat{\pi}_i = \pi_i \setminus \{\pi_{i,h} \in \pi_i : w_{\pi_{i,h},\psi_{i,h}} = \varepsilon\} \tag{6.36}$$

$$\hat{\psi}_i = \psi_i \setminus \{\psi_{i,h} \in \psi_i : w_{\pi_{i,h},\psi_{i,h}} = \varepsilon\} \quad (6.37)$$

$$\hat{m} = \sum_{i=0}^{l} \hat{\phi}_i \quad (6.38)$$

Namely, quantities $\hat{\phi}_i$, $\hat{\pi}_i$, $\hat{\psi}_i$, and $\hat{m}$ correspond to ϕ_i, π_i, ψ_i, and m respectively, after the removal of the ε words. The original realization $\mathbf{w}(\phi,\pi,\psi)$ of length m becomes the modified realization $\mathbf{w}(\hat{\phi},\hat{\pi},\hat{\psi})$ of length $\hat{m}$.

6.6.3 The CN based SLT log-linear model

The CN based SLT model $\Pr(\tilde{\mathbf{e}},\mathbf{a} \mid \mathcal{G})$ introduced in the search criterion (6.34) is defined as an extension of the SMT log-linear model presented in Chapter 5. As features functions for SMT are trained over a set of phrase pairs, which do not cope with ε-words, they have to be slightly changed.
Three feature functions exploit the modified quantities $(\hat{\phi},\hat{\pi},\hat{\psi})$ as follows:

$$h_1(\mathbf{s}) = h_1(\tilde{\mathbf{e}},\mathcal{G},\phi,\pi,\psi) = \log\Pr(\tilde{\mathbf{e}}) \quad (6.39)$$

$$h_2(\mathbf{s}) = h_2(\tilde{\mathbf{e}},\mathcal{G},\phi,\pi,\psi) = \log\Pr(\hat{\phi} \mid \tilde{\mathbf{e}}) \quad (6.40)$$

$$h_3(\mathbf{s}) = h_3(\tilde{\mathbf{e}},\mathcal{G},\phi,\pi,\psi) = \log\Pr(\mathbf{w}(\hat{\phi},\hat{\pi},\hat{\psi}) \mid \hat{\phi},\hat{\pi},\hat{\psi},\tilde{\mathbf{e}}) \quad (6.41)$$

The fourth feature function of the SMT model is related to the reordering of source phrases and is defined in terms of the distance between the first word of a source and the center of the previous one. In the case of the confusion networks, the real distance after the removal of ε-words) should be taken into account. As in general this distance is set only when the hypothesis is completed, an expected value is considered during the decoding. More formally we define the expected distance between the source phrases $\mathbf{w}(\phi_{i-1},\pi_{i-1},\psi_{i-1})$ and $\mathbf{w}(\phi_i,\pi_i,\psi_i)$ aligned with two consecutive target phrases $\tilde{\mathbf{e}}_{i-1}$ and $\tilde{\mathbf{e}}_i$ as the expected distance between the first column covered by a real word $\hat{\pi}_{i,1}$ and the

previous centroid $\hat{\bar{\pi}}_i$.

$$expdist(\pi_{i,1}, \bar{\pi}_i) = \sum_{t=\hat{\bar{\pi}}_i}^{\hat{\pi}_{i,1}} (1 - p_\varepsilon(t)) + p_\varepsilon(\hat{\pi}_{i,1}) \tag{6.42}$$

where $p_\varepsilon(t)$ is the probability of the ε-word in the t-th column of the CN if it exists, or equals to 0 otherwise. Last term of (6.42) is introduced because in position $\hat{\pi}_{i,1}$ a real word is surely covered. Notice that the indexes of the sum are inverted if $\hat{\pi}_{i,1} < \hat{\bar{\pi}}_i$. The lowest level sample-based distortion model (see Equation 4.16) becomes:

$$p(\pi_i \mid \phi, \bar{\pi}) = p_{=1}(expdist(\pi_{i,1}, \bar{\pi})) \prod_{k=2}^{\phi} \delta(\pi_{i,k} - \pi_{i,k-1} = 1) \tag{6.43}$$

Assuming this modification the fourth feature does not change:

$$h_4(\mathbf{s}) = h_4(\tilde{\mathbf{e}}, \mathcal{G}, \phi, \pi, \psi) = \log \Pr(\pi \mid \phi) \tag{6.44}$$

The fifth feature would model the real length of a realization of the CN through the following distribution:

$$\begin{aligned} h_5(\mathbf{s}) &= h_5(\tilde{\mathbf{e}}, \mathcal{G}, \phi, \pi, \psi) \\ &= \log \prod_{i=0}^{l} \prod_{h=1}^{\phi_i} \begin{cases} p_\varepsilon(\pi_{i,h}) & \text{if } w_{\pi_{i,h}, \psi_{i,h}} = \varepsilon \\ 1 - p_\varepsilon(\pi_{i,h}) & \text{otherwise} \end{cases} \\ &= \sum_{i=0}^{l} \log \prod_{h=1}^{\phi_i} \begin{cases} p_\varepsilon(\pi_{i,h}) & \text{if } w_{\pi_{i,h}, \psi_{i,h}} = \varepsilon \\ 1 - p_\varepsilon(\pi_{i,h}) & \text{otherwise} \end{cases} \end{aligned} \tag{6.45}$$

The last feature function consists of the posterior probability of a realization of the confusion network given the acoustic observation $\mathbf{o}$. In other words, it measures how probable the string $\mathbf{f} = \mathbf{w}(\phi, \pi, \psi)$ is within $\mathcal{G}$. By using the decomposition of $\Pr(\mathbf{f} \mid \mathcal{G})$ given in (6.5), and by remembering that each word of $\mathbf{f}$ corresponds to any word $\mathbf{w}(\phi_i, \pi_{i,h}, \psi_{i,h})), \exists\, 0 \leq i \leq l, 1 \leq h \leq d_i$, we obtain:

$$h_6(\mathbf{s}) = h_6(\tilde{\mathbf{e}}, \mathcal{G}, \phi, \pi, \psi)$$

$$
\begin{aligned}
&= \log \Pr(\mathbf{w}(\phi,\pi,\psi) \mid \mathbf{o}) \\
&= \log \prod_{i=0}^{l} \Pr(\mathbf{w}(\phi_i,\pi_i,\psi_i) \mid \mathbf{o},\pi_i) \\
&= \log \prod_{i=0}^{l} \prod_{h=1}^{\phi_i} \Pr(\mathbf{w}(\phi_i,\pi_{i,h},\psi_{i,h}) \mid \mathbf{o},\pi_{i,h}) \\
&= \log \prod_{i=0}^{l} \prod_{h=1}^{\phi_i} p_{\pi_{i,h},\psi_{i,h}} \\
&= \sum_{i=0}^{l} \log \prod_{h=1}^{\phi_i} p_{\pi_{i,h},\psi_{i,h}} \qquad (6.46)
\end{aligned}
$$

It is worth noticing that $\Pr(\mathbf{w}(\phi,\pi,\psi) \mid \mathbf{o})$, and hence $h_6(\mathbf{s})$ only depend on ϕ, π and ψ.

The contribution of each of the six features to the first step of the generative process is expressed by the following partial functions:

$$
\begin{aligned}
h_1(\mathbf{s};0) &= 0 && (6.47)\\
h_2(\mathbf{s};0) &= \log p(\hat{\phi}_0 \mid m - \hat{\phi}_0) && (6.48)\\
h_3(\mathbf{s};0) &= \log p(\mathbf{w}(\hat{\phi}_i,\hat{\pi}_i,\hat{\psi}_i) \mid \varepsilon) && (6.49)\\
h_4(\mathbf{s};0) &= -\log \phi_0! && (6.50)\\
h_5(\mathbf{s};0) &= \log \prod_{h=1}^{\phi_0} \begin{cases} p_\varepsilon(\pi_{0,h}) & \text{if } w_{\pi_{0,h},\psi_{0,h}} = \varepsilon \\ 1 - p_\varepsilon(\pi_{0,h}) & \text{otherwise} \end{cases} && (6.51)\\
h_6(\mathbf{s};0) &= \log \prod_{h=1}^{\phi_0} p_{\pi_{0,h},\psi_{0,h}} && (6.52)
\end{aligned}
$$

and the contribution to the i-th step by:

$$
\begin{aligned}
h_1(\mathbf{s};i) &= \log p(k_i) + \log p(\tilde{e}_i \mid \tilde{e}_{i-2},\tilde{e}_{i-1}) && (6.53)\\
h_2(\mathbf{s};i) &= \log p(\hat{\phi}_i \mid \tilde{e}_i) && (6.54)\\
h_3(\mathbf{s};i) &= \log p(\mathbf{w}(\hat{\phi}_i,\hat{\pi}_i,\hat{\psi}_i) \mid \hat{\phi}_i,\tilde{e}_i) && (6.55)\\
h_4(\mathbf{s};i) &= \log p(\pi_i \mid \phi_i,\bar{\pi}_{i-1}) && (6.56)
\end{aligned}
$$

$$h_5(\mathbf{s};i) = \log \prod_{h=1}^{\phi_i} \begin{cases} p_\varepsilon(\pi_{i,h}) & \text{if } w_{\pi_{i,h},\psi_{i,h}} = \varepsilon \\ 1 - p_\varepsilon(\pi_{i,h}) & \text{otherwise} \end{cases} \tag{6.57}$$

$$h_6(\mathbf{s};i) = \log \prod_{h=1}^{\phi_i} p_{\pi_{i,h},\psi_{i,h}} \tag{6.58}$$

The decomposition of the six features functions induces the definition of the following quantities:

$$R(\mathbf{s};\lambda) = R(\tilde{\mathbf{e}}, \mathcal{G}, \phi, \pi, \psi; \lambda) = \sum_{i=0}^{l} R(\mathbf{s};\lambda,i) \tag{6.59}$$

$$R(\mathbf{s};\lambda,i) = R(\tilde{\mathbf{e}}, \mathcal{G}, \phi, \pi, \psi; \lambda, i) = \sum_{r=1}^{6} \lambda_r h_r(\mathbf{s};i) \tag{6.60}$$

Hence, the complete SLT model $\Pr(\tilde{\mathbf{e}}, \mathbf{a} \mid \mathcal{G})$ is expressed as follows:

$$\Pr(\tilde{\mathbf{e}}, \mathbf{a} \mid \mathcal{G}) = \Pr(\tilde{\mathbf{e}}, \phi, \pi, \psi \mid \mathcal{G}) \tag{6.61}$$

$$= \frac{\exp R(\tilde{\mathbf{e}}, \mathcal{G}, \phi, \pi, \psi; \lambda)}{\sum_{\tilde{\mathbf{e}}'} \sum_{(\phi',\pi',\psi') \in \mathcal{A}(\mathcal{G},\tilde{\mathbf{e}}')} \exp R(\tilde{\mathbf{e}}', \mathcal{G}, \phi', \pi', \psi'; \lambda)} \tag{6.62}$$

$$\propto \exp R(\tilde{\mathbf{e}}, \mathcal{G}, \phi, \pi, \psi; \lambda) = \exp R(\mathbf{s};\lambda) \tag{6.63}$$

and the search criterion (6.34) becomes:

$$\mathbf{s}^* \approx \arg\max_{\mathbf{s}} R(\mathbf{s};\lambda) \tag{6.64}$$

which directly provides also the best translation $\tilde{\mathbf{e}}^*$.

6.6.4 Search Problem

By looking at formulas (6.47-6.58), we point out that the state of the partial solution $(\mathbf{s}, i) = (\tilde{\mathbf{e}}, \mathcal{G}, \phi, \pi, \psi; i)$ is $[\mathbf{s}; i] = (\mathcal{C}, \bar{\pi}_i, \tilde{e}_i, \tilde{e}_{i-1})$, where $\mathcal{C} = \bigcup_{t=0,\ldots,i} \pi_t$. Notice that the state of a partial solution is defined exactly in the same way as in the SMT model, because the extension score at step i depends on ψ_i, but not

on the previous ones. In particular, this means that two partial solutions can be recombined even if they exploit different realizations of the CN.
The first step of the generative process produces only partial solutions $(\mathbf{s},0)$ of length 0 with state $(\pi_0,\bar{\pi}_0,\varepsilon,\varepsilon)$. Hence, all partial solutions of length 0 sharing a given $[\mathbf{s}]=(\pi_0,\bar{\pi}_0,\varepsilon,\varepsilon)$ have a variable ψ_0. This means that:

$$\begin{aligned} T_0([\mathbf{s}];0) &= T_0(\pi_0,\bar{\pi}_0,\varepsilon,\varepsilon;0) \\ &= \max_{\psi_0} R(\tilde{\mathbf{e}},\mathcal{G},\phi,\pi,\psi;\lambda,0) \end{aligned} \tag{6.65}$$

In the generic i-th step of the generative process the set $Pred([\mathbf{s}])$ of a partial solution $(\mathbf{s},i)=(\tilde{\mathbf{e}},\mathcal{G},\phi,\pi;i)$ of state $[\mathbf{s}]=(\mathcal{C},\bar{\pi},\tilde{e}_i,\tilde{e}_{i-1})$ contains states $[\mathbf{s}']=(\mathcal{C}\setminus\pi_i,\bar{\pi}',\tilde{e}_{i-1},\tilde{e}'')$. The corresponding expansion score is:

$$S([\mathbf{s}'],[\mathbf{s}])=S(\mathcal{C},\pi_i,\bar{\pi}_i,\tilde{e}_i,\tilde{e}_{i-1},\tilde{e}_{i-2})=\max_{\psi_i} R(\tilde{\mathbf{e}},\mathcal{G},\phi,\pi,\psi;\lambda,i) \tag{6.66}$$

Notice that ϕ_i is univocally determined by π_i.
The maximization in equation (5.42) maximization over generic $\tilde{e}_{i-2}$, $\emptyset\subseteq\pi_i\subseteq\mathcal{C}$, ψ_i, and $\bar{\pi}_{i-1}$:

$$\begin{aligned} T((\mathcal{C},\bar{\pi}_i,\tilde{e}_i,\tilde{e}_{i-1});\lambda,i) = \max_{\tilde{e}_{i-2},\emptyset\subseteq\pi_i\subseteq\mathcal{C},\psi_i,\bar{\pi}_{i-1}} & \\ T((\mathcal{C}\setminus\pi_i,\bar{\pi}_{i-1},\tilde{e}_{i-1},\tilde{e}_{i-2});\lambda,i-1) & \\ R(\tilde{\mathbf{e}},\mathcal{G},\phi,\pi,\psi;\lambda,i) & \end{aligned} \tag{6.67}$$

Besides the reordering constraint and the probability cutoff, introduced in Section 4.3.2, a third method can be applied to limit the number of theories to generate, which limits the depths of the columns of the input CN $\mathcal{G}$.

- *Confusion Network cutoff*: less input words are considered in the source CN by removing terms $w_{j,k}$ with posterior probabilities $p_{j,k}$ are below a given threshold. Eventually, a whole column is removed if it contains no words or only the ε-word.

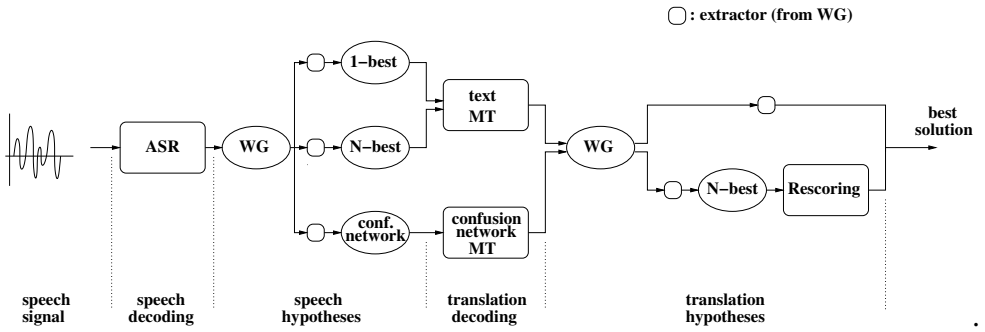

Figure 6.3: The ITC-irst Spoken Language Translation System

6.6.5 Analysys of the complexity

We have just shown that the search algorithm for the CN-based SLT model differs from that for the SMT model only in the way quantities T and S are computed. In fact, the maximum operations in equations (6.65) and (6.67) are performed over one more variable, ψ_0 and ψ_i, respectively. Henceforth, the complexity for the computation of the single subproblem increases at most of a factor $d_{max}^{\phi_{max}}$, where d_{max} is the largest depth of the CN. As the number of subproblems is equivalent due to the identity of the state definition, the complexity of the algorithm is $O\left(2^m\, m^3\, \phi_{max}\, d_{max}^{\phi_{max}} \left(\begin{array}{c} m \\ \phi_{max} \end{array}\right) |\mathcal{E}|^3\right)$.

6.7 The ITC-irst SLT system

Figure 6.3 illustrates the speech translation system currently developed at ITC-irst [6], which can be virtually divided into two parts. In the left-hand side, beginning from the speech signal of the utterance, the ASR system (see Section 6.2.1) produces a word graph that contains alternative recognition hypotheses. By using the word graph, we can extract an N-best list, eventually only the best transcription, and pass it to the N-best MT decoder (see Section 6.4). Otherwise, a confusion network can also be built from the word graph with a

dominant property that it has a more compact representation and a lower word error rate than the former. In this case the CN-based MT decoder presented in Section 6.6 is used. Similarly, in the right-hand side, output of machine translation is again a word graph, a compact representing of the translation hypotheses in the target language. Clearly, if we are just interested in the translation result, the best translation hypothesis can be extracted directly from the word graph. Additionally, the possibility of having word graphs and *N*-best list outputs allows to optimize parameters of the MT and ASR systems or rescore translation hypotheses with deeper and more extensive knowledge sources. Preprocessing and postprocessing, even if not reported in Figure 6.3, are performed as in the text MT system (see Section 4.5).

6.8 Experiments

In this Section the *N*-best-based and CN-based approaches presented in the previous Sections are compared. As a reference we also translate the manual transcription, which is the upper bound for the SLT task.
Experiments regard both the translation quality measured with `BLEU`, `NIST` and `WER`, and the efficiency evaluated in terms of the average number of hypotheses, `Avg#Th`, generated during the decoding (see Appendix B.2).

6.8.1 Training and Testing Data

Evaluation of the *N*-best-based and CN-based SLT systems were performed over the same benchmark described in Section 4.6.2. Table 6.1 summaries statistics of test data sets available for the ASR system.
The language model of the ASR system has been trained over the Italian part of BTEC, and has a perplexity of 60 and 44 over the test sets Q1 and Q2, respectively. The acoustic model has been trained over a 130h corpus of broadcast news. Model weights of the ASR system are optimized over a development

	#sent.	W	|V|	#spk	speech	WER	OOV	PP
Q1	3006	23512	2768	17 (8f+9m)	3h:25m	21.9	4.3%	60
Q2	506	2985	940	10 (5f+5m)	29m	23.1	3.4%	44

Table 6.1: Audio statistics of the two test sets, Q1 and Q2. Number of sentences, running words, dictionary size, number of speakers, audio length, word error rate of the 1-best list, out-of-vocabulary rate, and language mdoel perplexity are reported.

set of 500 speech utterances through the Minimum Error Training method described in Section 5.3.1. After the weight estimation, the ASR system achieves a WER of 21.9% and 23.1% over the Q1 and Q2, respectively.

It is worth remarking that speech transcriptions and corresponding translations are produced without punctuation and caseing information; hence, training data for MT are preprocessed accordingly.

6.8.2 Relationship between recognition and translation quality

First of all we were interested in finding if recognition and translation quality are correlated. As the ASR system is not perfect, the best transcription provided by the ASR system, i.e. that with the highest score, does not achieve, necessarily, the best recognition accuracy, i.e. the lowest WER. An oracle can extract the transcription which minimizes WER from a set of alternatives. Obviously, the larger this set, the better quality of the oracle transcriptions.

Oracle transcriptions were extracted from sets of increasing sizes, which achieved lower WER, and translated. In particular, oracles were extracted from N-best lists of different depth and confusion networks with different pruning. Figure 6.4 shows the almost linear correlation between recognition errors (WER) and translation quality (BLEU) that were observed.

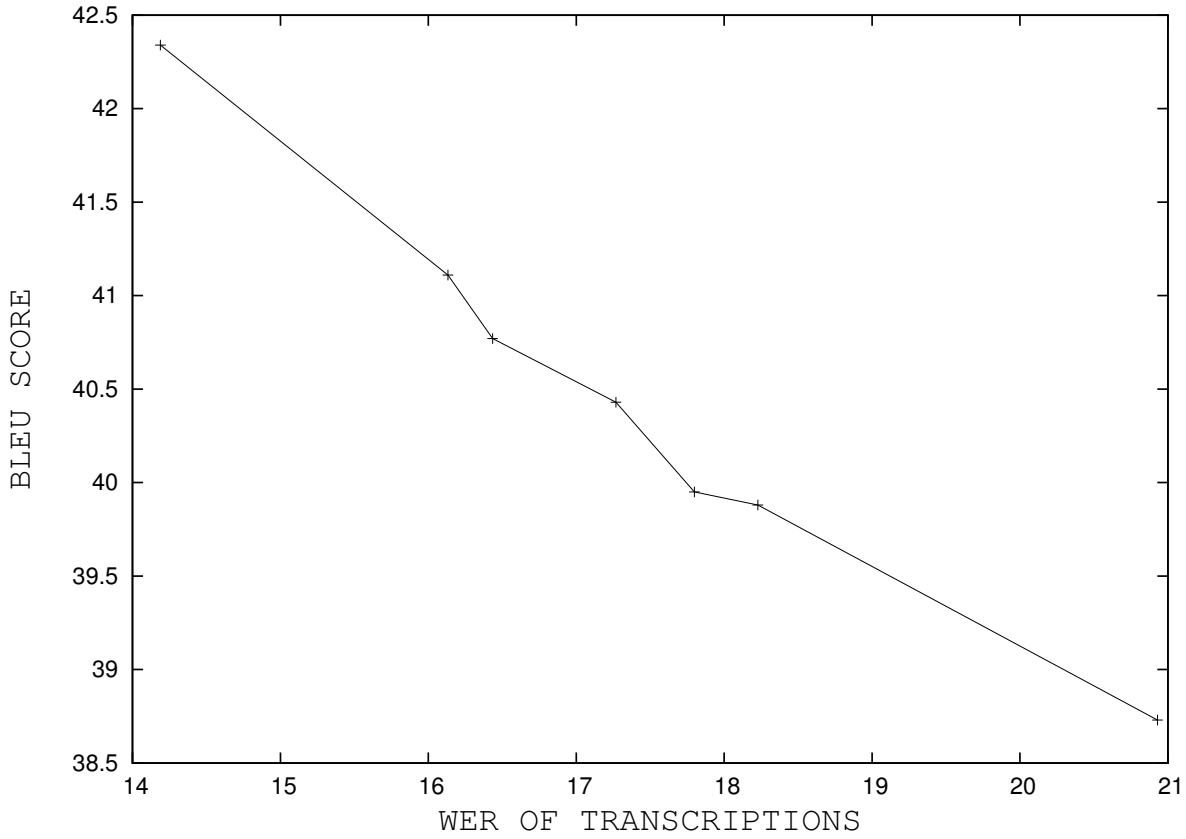

Figure 6.4: Correlation between recognition accuracy and translation quality. Oracle transcriptions of increasing accuracy are translated and evaluated with `BLEU` score.

6.8.3 Comparison of the SLT systems

The comparison of the two systems, namely the *N*-best and the CN decoders, was performed from the point of view of translation quality and decoding efficiency. Both systems were set with uniform model weights, and produced word graphs including all translation hypotheses generated during the decoding. Then, best translations were extracted after rescoring with optimal weights.

The *N*-best MT system was applied to the `1-best`, `15-best`, `10-best`, `20-best`, `50-best` and `100-best` ASR transcriptions. Moreover, this system was also used to translate the `human` transcriptions of the test sets. The CN-based MT system instead translated from confusion networks which were pruned according to the criterion presented in Section 6.6.4. On the average, each confusion

	BLEU	NIST	MWER	Avg#Th	Avg#Tr
human	53.0 (51.9-54.2)	9.70 (9.58-9.83)	32.71 (31.48-33.91)	41K	1180
1-best	39.9 (38.7-41.0)	8.03 (7.91-8.18)	46.0 (44.4-47.5)	38K	1215
5-best	40.9 (39.8-42.1)	8.18 (8.05-8.31)	44.5 (43.0-46.0)	218K	5929
10-best	41.0 (39.9-42.3)	8.20 (8.07-8.33)	44.3 (42.8-45.7)	445K	11610
20-best	41.2 (40.1-42.4)	8.21 (8.08-8.35)	44.2 (42.7-45.7)	900K	22547
50-best	41.2 (40.1-42.4)	8.22 (8.09-8.35)	44.1 (42.6-45.6)	2255K	56690
100-best	41.3 (40.2-42.5)	8.22 (8.09-8.36)	44.1 (42.6-45.5)	4501K	113991
CN	40.1 (39.0-41.2)	8.01 (7.88-8.14)	45.8 (44.2-47.2)	319K	4350

Table 6.2: Comparison of the SLT systems on the test Q1.

	BLEU	NIST	MWER
human	64.9	10.6	28.9
1-best	54.3	9.24	37.6
5-best	56.5	9.55	35.6
10-best	56.7	9.58	35.5
20-best	57.0	9.59	35.4
50-best	57.0	9.59	35.3
100-best	56.8	9.59	35.6
CN	56.08	8.754	37.91

Table 6.3: Comparison of the SLT systems on the test Q2.

network contains 54K transcription hypotheses.

Performance on test sets Q1 and Q2 are reported in Table 6.2 and 6.3. Moreover, the average number of hypotheses computed during the decoding (Avg#Th) and the average number of different translation alternatives (Avg#Tr) stored in the word-graphs are given for Q1.

First of all, we observe that, by moving from human to automatic transcriptions, translation quality decreases by about 25% in terms of BLEU score (from 53.0 to 39.9 on Q1).

By looking at performance of the *N*-best SLT system, we notice that the major improvement with respect to the 1-best is obtained by simply consider 5 alter-

natives, and differences become slightly significant above $N = 10$. Increasing the number of transcription hypotheses does not improve performance, but only affects the computation effort.
As concerns the CN-based system, we observe that it performs similarly to the 1-best system. By looking at the computation level, the CN decoder competes with the 5-best and 10-best systems, but performance are lower, although not significantly.
We believe that the not very promising result of the CN-based system is due to the method of generating the confusion network itself. Main advantage of the CN-based system consists in its decoding algorithm which permits to consider during the search a much larger number of transcription hypotheses (54K) than the N-best-based system. But as shown in the matrix representation depicted in Figure 6.2, posterior probabilities of words are very sharp. This means that almost all probability mass of the CN is concentrated in very few hypotheses, among the 54K alternatives. Hence, any translation of the remaining hypotheses achieves a global probability very close to 0, due to feature h_6 (see Section 6.6).

6.8.4 Potential quality of the SLT approaches.

The potential quality of the two SLT approaches can be measured by applying the previous oracle strategy to the produced translation alternatives. Hence, we picked translations with the lowest `WER` from a set of M-best alternatives.
Figure 6.5 plots the `WER` achieved by the considered SLT systems with increasing M. Notice that real performance of a system is obtained when $M = 1$, while, on the opposite, a full oracle approach is given if M has the largest value.
The curve corresponding to CN-based system decreases faster than the curve of the N-best systems when M is larger than 10. This means that the CN-based system finds a larger number of "good" translations among a smaller number of alternatives.
Interestingly, the CN-based system achieves the same performance (`WER 31.00`)

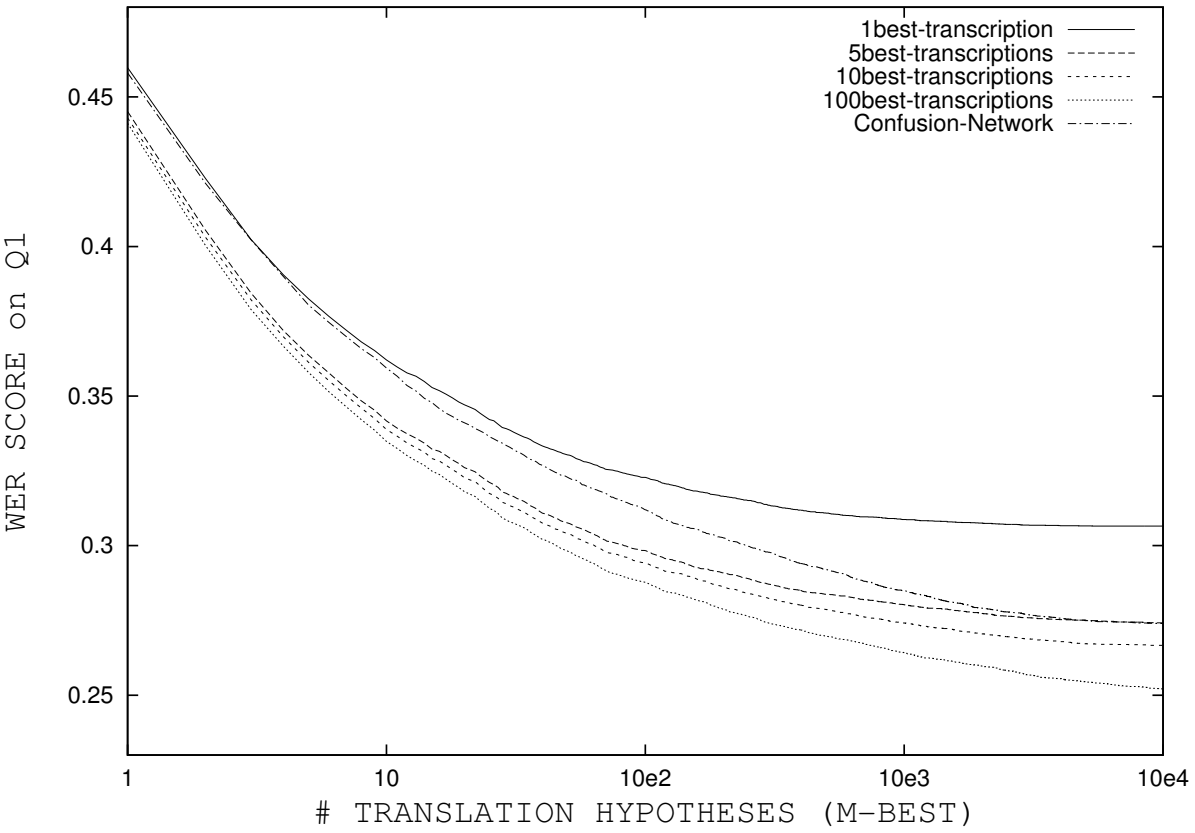

Figure 6.5: Correlation between recognition accuracy and translation quality. Oracle transcriptions of increasing accuracy are translated and evaluated with BLEU score.

of the 1-best system within a significant lower number of translation hypotheses ($M = 100$ vs $M = 1000$). Morever, at the same level $M = 1000$ and $M = 10000$, the CN-based system outperforms the 1-best system of about 8% and 10% respectively.

Chapter 7

Conclusions

This dissertation has presented original work in three area of Machine Translation: Cross-Language Retrieval, Text Translation, and Spoken Language Translation. In particular, new statistical models and innovative search algorithm have been proposed. Main results of my work are now briefly summarized.

7.1 Cross-Language Information Retrieval

The following results emerged from my research on Cross-Language Information Retrieval and participation in several CLEF evaluation campaigns.

- The statistical LM approach well compares with the Okapi model, and results very competitive on long topics or after query expansion. Moreover, consistent improvement in performance over both methods was achieved by combining the Okapi and LM scores after some normalization.
- Comparing CLIR models results quite difficult. As a matter of fact, retrieval performance seems very sensitive to translation quality, which however depends on the coverage of the available dictionaries and on the generation of correct word stems and base-forms. Retrieval performance measured by using our translation model and a commercial translation system showed, over a set of 140 queries, many large fluctuations. In fact, such

high variability did not permit to rank the approaches in a statistically significant way, at least on the available sets of queries. From our point of view, this also means that, for the sake of IR, our statistical translation model, which is quite simple to implement, did not perform worse than a state-of-the-art commercial translation engine, which was developed over several decades.

- Qualitative analysis of results suggests that improvements in CLIR should be pursued in two main directions: by developing better statistical CLIR models (see below), and by augmenting coverage of bilingual dictionaries. On the other hand, recent experiments showed that text preprocessing based on morpho-syntactic analysis is not superior than basic word stemming. This widens the applicability of the proposed CLIR approach to other language pairs for which bilingual dictionaries are available. Further experiments [8] were carried out on a cross-language spoken document retrieval track, with spoken documents in English and topics in French, German, Italian, and Spanish. Promising results were achieved by only using publicly available dictionaries and stemming algorithms.

- Finally, it is well known that blind relevance feedback is determinant step which boosts retrieval performance, especially for short queries. However, up to now, not enough effort has been devoted to embed BRF into a statistically sound framework. Besides this theoretical issue, it would be useful investigate how BRF could be specifically devised for CLIR, for instance, to improve quality of translations.

7.2 Text Translation

Initial efforts were devoted to develop a statistical MT system from scratch. In particular, two issues related to a popular word-based statistical MT model by

[12] were addressed: the derivation of an efficient decoding algorithm, and the extension of the model to include stochastic dependencies beyond single words.

- An original dynamic programming formulation of the decoding problem was derived proposed for the so-called Model 4, which directly derives from the search optimization criterion. Moreover, an approximate decoding algorithm was implemented which applies theory pruning and word-reordering constraints in order to keep decoding time under control.

- Following a recent trend in MT to exploit statistics at level of phrases, three extensions of Model 4 have been discussed, which model phrase-level probabilities, respectively, (i) by integrating word-based probabilities, (ii) through empirical measures on a sample of phrases, (iii) by combining the two previous methods. Remarkably, all the resulting models do not require to modify the decoding algorithm for Model 4. Practically, they just require augmenting the target dictionary and the original parameters of Model 4. Given a parallel corpus for training, a sample of phrases is simply extracted by exploiting bi-directional alignments computed with word-based translation models [60].

- Extensive translation experiments on a tourism domain, with translation directions Chinese-English and Italian-English, showed that the presented phrase-based models are superior to Model 4. In particular, most robustness against data sparseness is shown by the model combining word-based and sample-based statistics. Moreover, the trade-off between memory and time consumption vs. performance was analyzed with respect to the maximum allowed phrase length. Our conclusion that phrases of up-to three words provide a reasonable trade-off agrees with the outcome of [41].

7.3 Spoken Language Translation

As concerns the Spoken Language Translation task, we have proposed a new statistical model for integrating ASR and MT systems.

- We extend the statistical model for text translation to embed acoustic evidence. Two systems have been developed: the first exploits a list of N-best transcription hypotheses produced by the ASR system, while the second is based on a particular word graph, a confusion network, which approximates the word graph generated by the ASR system. In both cases, a log-linear model is defined, which combines a set of real-valued features.

  Moreover, the search algorithm implemented for the confusion network system is obtained by properly extending th phrase-based decoder.

- N-best-based systems (with $N \geq 10$) significantly ouperform the 1-best decoder, but the decoding time, which is almost linear in N, becomes critical.

  Instead, the CN-based system has the main advantage of translating a large set of hypotheses very fast. As concerns translation quality we found that, the CN-based system does not perform significantly worse than the N-best systems, apart from the case $N = 100$ which, however, requires a much larger computational effort.

- We think that performance of the CN-based system are strongly affected by a characteristic of the confusion networks, we worked with. Even if they contain a lot of transcription alternatives, almost all probability mass is concentrated in very few of them. Practically, all remaining hypotheses are pruned during the decoding process. We expect that smoothing the confusion network probabilities could significantly improve translation quality without impacting over decoding time.

Appendix A

Word Graphs

A word graph (WG) is a directed, acyclic, weighted, labeled graph with distinct start and end nodes. It is a quadruple $G = (V, E, I, F)$ where $V = \{v_1, ..., v_N\}$ is a set of nodes, $E = \{e_1, ..., e_M\}$ is a set of edges, and $I, F \in V$ are the start and end nodes. By the default, $I = v_1$ and $F = v_N$. An edge $e = (v_i, v_j)$ connects the starting node $v_i \in V$ to the ending node $v_j \in V$, and is labelled with features related to the application.

For instance, edge labels of the word-graphs produced by the ITC-irst ASR system consists of a recognized word hypothesis, its starting and ending time, and the acoustic and language model scores. Instead, the ITC-irst SMT decoder produces word-graphs, whose edges are labelled with the target phrases, the alignments, besides the scores.

A.1 Word Graph decoding

WG decoding is the process of finding the best sentence and the $N-$best sentences through the WG. Moreover, an approximation of the WG can be obtained which is more compact and suitable to feed the SLT system described in Section 6.6. More details can be found in [67].

A.1.1 1-best Word Graph decoding

Finding the best sentence in the word graph is equivalent to the shortest path problem in graph theory [18]. Dynamic programming permits to solve this problem in a very efficient way.

A.1.2 *N*-best decoding

The problem of finding the $N-$shortest paths of a weighted directed graph is a well-studied problem in computer science [22]. In MT and in ASR the problem is slightly different, because it is often desirable to determine not just the *N*-best word sequences, but the *N*-best distinct word sequences. An efficient algorithm proposed in [81] was implemented, which perform an exact search without any approximation.

A.1.3 Confusion Network

A WG generated by an ASR system can be compacted into a so-called *confusion network* (CN) by means of an algorithm proposed by [46]. A CN, which is a acyclic directed word-graph, is linear in the sense that every path from the start to the end node has to pass through all nodes and, consequently, all paths have the same length. It is worth noticing that a CN contains more transcription hypotheses than the original WG.
Figures A.1, A.2, and A.3 show the word graph generated from the ITC-irst ASR system, the corresponding confusion network, and a list of *N*-best transcription hypotheses.

A.2 Word Graph evaluation methods

The quality of a WG is usually related to its size and the *graph word error rate* (GWER). The *word graph density* (WGD), defined as the total number of edges di-

vided by the number of words of the reference, is a widely used criterion for measuring the size. The GWER is computed by determining which path within the WG minimizes the WER with respect to the reference sentence (see Appendix B.2). Practically, computation of GWER also supplies this best sentence hypothesis, which corresponds to the oracle response.

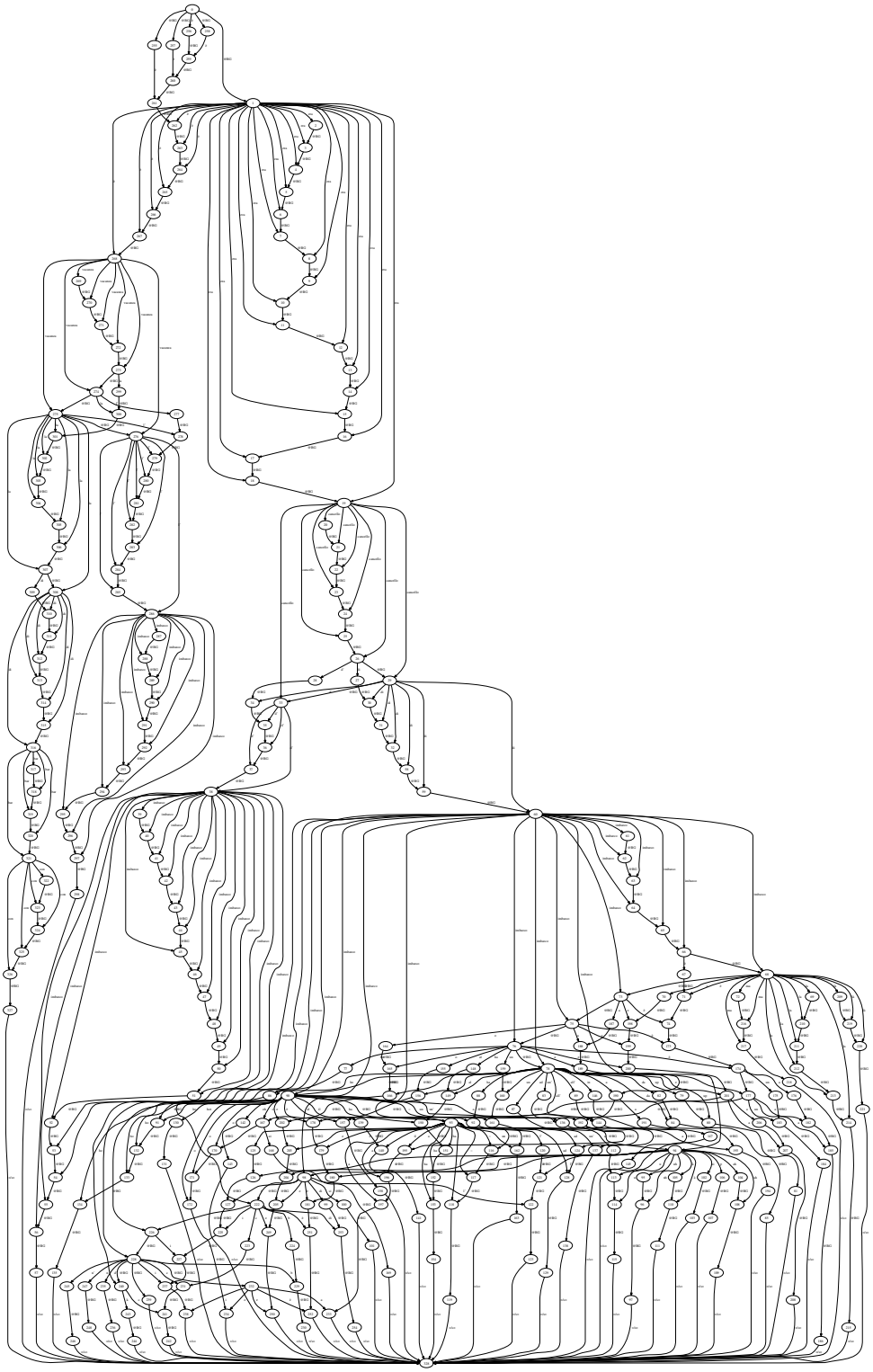

Figure A.1: Example of the word graph generated from the ITC-irst ASR system.

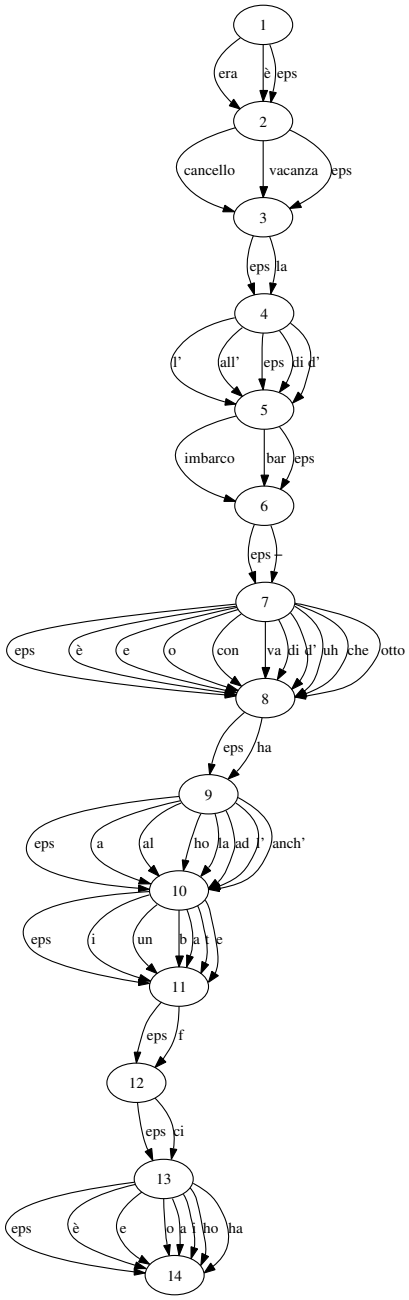

Figure A.2: Example of a confusion netword extracted from the word graph.

1	era cancello di imbarco
2	era cancello d' imbarco
3	era cancello di imbarco
...	...
8	è vacanza l' imbarco
9	era cancello di imbarco o
...	...
14	era cancello di imbarco la
15	era cancello di imbarco i
16	è vacanza la di bar con
...	...
human	dove è il cancello d' imbarco
gold reference	where is the boarding gate

Figure A.3: *N*-best transcriptions extracted from a word graph.

Appendix B

Evaluation Measures

B.1 Information Retrieval

Performance of an Information Retrieval (IR) system is usually measured in terms of *mean average precision* `mAvPr`.
Given the document ranking provided against a given query q, let $r_1 \leq \ldots \leq r_k$ be the ranks of the retrieved relevant documents. The `AvPr` for q is defined as the average of the precision values achieved at all recall points, i.e.:

$$\texttt{AvPr} = 100 \times \frac{1}{k} \sum_{i=1}^{k} \frac{i}{r_i} \tag{B.1}$$

The `mAvPr` of a set of queries corresponds to the mean of the corresponding query `AvPr` values.
An other widely used evaluation score is the *F1-measure* combining two orthogonal metrics, namely the *recall*, i.e. the ratio between the relevant documents returned by the system and the total number of relevant documents in the collection, and the *precision*, i.e. the ratio between the relevant document returned and the total number of document returned, as follows: $\texttt{F1} = \frac{2 \times precision \times recall}{precision + recall}$.
Performance difference between two IR system is considered statistically significant if the paired sign test [37] succeeds. In particular, the test is applied to paired average precision measures of single queries, by testing for a median difference of zero.

B.2 Machine Translation

Translation quality is evaluated by means of three well established automatic measures:

- Word Error Rate (WER) is the *edit* (or *Levenshtein*) distance [45], which computes the minimum number of substitution, insertion and deletion operations that have to be performed to convert the hypothesis into the reference sentence, divided by the length of the latter. If the test set consists of more sentences the global edit distance and the global reference length are taken into account, and successively divided. This performance criterion is widely used in ASR. Hence, ideally, the lower the score the better the translation.

- Multi-reference WER (MWER) is an extension of the WER in the case more references exist for each test sentence. For each hypothesis the edit distance to the most similar reference sentence is computed [55].

- BLEU score [62] corresponds to the geometric average of the precision measures of the n-grams, with n from 1 to 4, of the system output against the n-grams in all the references. As recall is not considered, a penalty for too short sentences is added. Unlike WER, BLEU score measures translation accuracy; hence, larger the values better the quality.

- NIST score [33] is a variant of BLEU, which computes the arithmetic average of the precisions of n-grams, with n from 1 to 5, and gives a higher penalty to very short sentences. The worst NIST score is zero while the best one is a positive number depending on the length of the test set.

As reported in [44, 62], all above measures correlate quite well with human judgments, especially if they are applied to compare different versions of the same system.

Automatic scores are really useful if they permits to decide whether difference between performances of two systems is significant. This can be achieved either by using large, but expensive, test suites or by applying an appropriate tool, the so-called *bootstrapping method* [93], developed in statistical testing theory. This method provides a confidence interval for a specific metric. Given the confidence intervals of two systems, a test of equality of the means of two normal distributions is applied, and a confidence level α=0.05 is considered.

A fair evaluation of MT decoder should take into account time and memory consumption, too. Since the ITC-irst system features a parallel decoding on a cluster of PCs, the elapsed time for translation could not be used as it significantly varies according with the workload and power of the engaged CPUs. A more stable measure has been adopted, namely the average number of translation hypotheses (`Avg#Th`) generated during the search algorithm. Figure B.1 shows the almost perfect correlation between the `Avg#Th` and the decoding time, which includes time for generating, scoring, recombining and pruning theories, for many runs under controlled conditions.

Memory consumption takes into account storage for model parameters and generated theories; the former is strictly related to the number of translation pairs contained in the bilingual dictionary, the latter is proportional to the number of generated theories.

Henceforth, when a time-memory based comparison is required, the average number of generated theories (`Avg#Th`) and the size of the used bilingual dictionary are used.

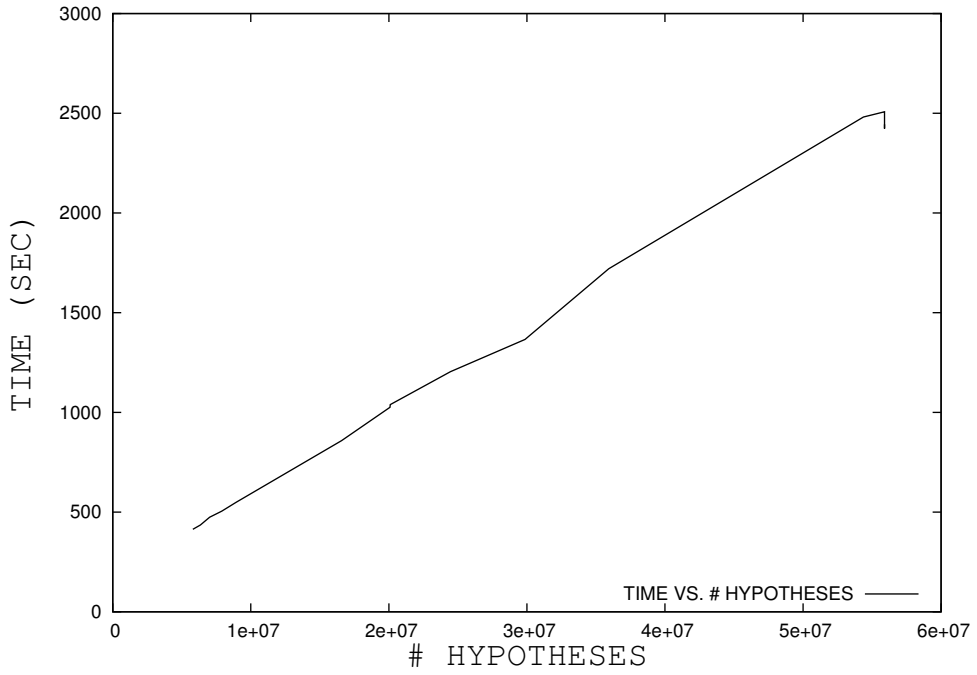

Figure B.1: Processing time as function of the generated hypotheses (`Avg#Th`).

Bibliography

[1] G. Antoniol, F. Brugnara, M. Cettolo, and M. Federico. Language model representations for beam-search decoding. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 588–591, Detroit, MI, 1995.

[2] L. Ballesteros and W. B. Croft. Resolving ambiguity for cross-language retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 64–71, Melbourne, Australia, 1998.

[3] A. Berger and J. D. Lafferty. Information retrieval as statistical translation. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 222–229, Berkeley, CA, 1999.

[4] A. Berger, S. A. D. Pietra, and V. J. D. Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[5] N. Bertoldi. Studio ed implementazione di modelli statistici per il problema del pos tagging. Master's thesis, Faculty of Mathematics, University of Trento, 2000.

[6] N. Bertoldi, R. Cattoni, and M. F. Mauro Cettolo. The itc-irst statistical machine translation system for IWSLT-2004. In *Proceedings of the 1st*

International Workshop on Spoken Language Translation, pages 51–58, Kyoto, Japan, 2004.

[7] N. Bertoldi and M. Federico. ITC-irst at CLEF 2000: Italian monolingual track. In C. Peters, editor, *Cross-Language Information Retrieval and Evaluation*, volume 2069 of *Lecture Notes in Computer Science*, pages 261–272, Heidelberg, Germany, 2001. Springer Verlag.

[8] N. Bertoldi and M. Federico. ITC-irst at CLEF-2003: Cross-Language Spoken Document Retrieval. In C. Peters, M. Braschler, J. Gonzales, and M. Kluck, editors, *Comparative Evaluation of Multilingual Information Access Systems*, volume 3237 of *Lecture Notes in Computer Science*, Heidelberg, Germany, 2003. Springer Verlag.

[9] P. Beyerlein. Discriminative model combination. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*, pages 238–245, Santa Barbara, CA, 1997.

[10] P. Beyerlein, X. L. Aubert, R. Haeb-Umbach, M. Harris, D. Klakow, A. Wendemuth, S. Molau, M. Pitz, and A. Sixtus. The Philips/RWTH system for the transcription of broadcast news. In *Proceedings of the 6th European Conference on Speech Communication and Technology*, pages 647–650, Budapest, Hungary, 1999.

[11] P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. C. Lai, and R. L. Mercer. Method and system for natural language translation. United States Patent 5,477,451, 1995.

[12] P. F. Brown, V. S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–312, 1993.

[13] R. Brown. Example-based machine translation in the pangloss system. In *Proceedings of the Sixteenth International Conference on Computational Linguistics*, volume 1, pages 169–174, Copenhagen, Denmark, 1996.

[14] F. Brugnara, M. Cettolo, M. Federico, and D. Giuliani. Advances in automatic transcription of italian broadcast news. In *Proceedings of the International Conference of Spoken Language Processing*, volume II, pages 660–663, Beijing, China, 2000.

[15] F. Brugnara, M. Cettolo, M. Federico, and D. Giuliani. A baseline for the transcription of Italian broadcast news. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Istanbul, Turkey, 2000.

[16] F. Brugnara and M. Federico. Dynamic language models for interactive speech applications. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, pages 2751–2754, Rhodes, Greece, 1997.

[17] F. Casacuberta, H. Ney, F. J. Och, E. Vidal, J. M. Vilar, S. B. adn I. Garcia-Varea, D. Llorens, C. Martinez, S. Molau, F. Nevado, M. Pastor, D. Pico, A. Sanchis, and C. Tillmann. Some approaches to statistical and finite-state speech-to-speech translation. *Computer Speech and Language*, 18:25–47, 2004.

[18] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press/ McGraw-Hill Book Company, 2001. Second edition.

[19] J. N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43(5):1470–1480, 1972.

[20] S. A. Della Pietra, V. J. Della Pietra, and J. Lafferty. Inducing features of random fields. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-19(4):380–393, 1997.

[21] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, B*, 39:1–38, 1977.

[22] D. Eppstein. Finding the *k* shortest paths. *SIAM Journal of Computing*, 28(2):652–673, 1998.

[23] M. Federico. A System for the Retrieval of Italian Broadcast News. *Speech Communication*, 32(1-2):37–47, 2000.

[24] M. Federico and N. Bertoldi. Statistical cross-language information retrieval using n-best query translations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 167–174, Tampere, Finland, 2002.

[25] M. Federico and R. De Mori. Language modelling. In R. D. Mori, editor, *Spoken Dialogues with Computers*, chapter 7, pages 199–230. Academy Press, London, UK, 1998.

[26] W. B. Frakes and R. Baeza-Yates, editors. *Information Retrieval: Data Structures and Algorithms*. Prentice Hall, Englewood Cliffs, NJ, 1992.

[27] J. Gao, J.-Y. Nie, E. Xun, J. Zhang, M. Zhou, and C. Huang. Improving query translation for cross-language information retrieval using statistical models. In *Proceedings of the 24th annual international ACM SIGIR Conference on Research and development in information retrieval*, pages 96–104, New Orleans, LA, 2001.

[28] J. Gao, M. Zhou, J.-Y. Nie, H. He, and W. Chen. Resolving query translation ambiguity using a decaying co-occurrence model and syntactic de-

pendence relations. In *Proceedings of the 25th annual international ACM SIGIR Conference on Research and development in information retrieval*, pages 183–190, Tampere, Finland, 2002.

[29] U. Germann, M. Jahr, K. Knight, D. Marcu, and K. Yamada. Fast decoding and optimal decoding for machine translation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 228–335, Toulouse, France, 2001.

[30] G. Grefenstette, editor. *The Problem of Cross-Language Retrieval Information Retrieval*, Boston, MA, 1998. Kluwer Academic.

[31] D. Hiemstra and F. de Jong. Disambiguation strategies for cross-language information retrieval. In *Proceedings of the 3rd European Conference on Research and Advanced Technology for Digital Libraries*, pages 274–293, Paris, France, 1999.

[32] http://world.altavista.com.

[33] http://www.nist.gov/speech/tests/mt.

[34] D. Hull and G. Grefenstette. Querying across languages: A dictionary-based approach to multilingual information retrieval. In *Proceedings of the 19th ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 49–57, New York, 1996.

[35] W. J. Hutchins. *Machine Translation: Past, present, future*. Academic press, 1986.

[36] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press Cambridge, Massachusetts, London, England, 1997.

[37] R. A. Johnson and D. W. Wichern, editors. *Applied Multivariate Statistical Analysis*. Prentice Hall, Englewood Cliffs, NJ, 1992.

[38] S. Johnson, P. Jourlin, K. S. Jones, and P. Woodland. Spoken document retrieval for TREC-8 at Cambridge University. In *Proceedings of the 8th Text REtrieval Conference*, pages 197–206, Gaithersburg, MD, 1999.

[39] D. Klakow. Log-linear interpolation of language models. In *Proceedings of the International Conference of Spoken Language Processing*, pages 1695–1698, Sidney, Australia, 1998.

[40] K. Knight. Decoding complexity in word replacement translation models. *Computational Linguistics*, 25(4):607–615, 1999.

[41] P. Koehn, F. J. Och, and D. Marcu. Statistical phrase-based translation. In *Proceedings of the Human Language Technology and North American ACL Conference*, pages 127–133, Edmonton, Canada, 2003.

[42] K. L. Kwok. English-chinese cross-language retrieval based on as translation package. In *Proceedings of VII Machine Translation Summit. Workshop: Machine Translation for Cross-Language Information Retrieval*, pages 8–13, Singapore, 1999.

[43] K. L. Kwok, L. Grunfeld, N. Dinstl, and M. Chan. Trec-9 cross language, web and question-answering track - experiments using PIRCS. In *Proceedings of the 8th Text Retrieval Conference TREC-9*, pages 419–429, 2000.

[44] G. Leusch, N. Ueffing, and H. Ney. A novel string-to-string distance measure with applications to machine translation evaluation. In *Proceedings of the Machine Translation Summit IX*, pages 240–247, New Orleans, LA, 2003.

[45] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals (in Russian). *Problemy Peredachi Informatsii*, 1(1):12–25,

1965. English translation in Problems of Information Transmission, 1 (No. 1, 1965), 8–17.

[46] L. Mangu, E. Brill, and A. Stolcke. Finding consensus among words: Lattice-based word error minimization. In *Proceedings ISCA European Conference on Speech Communication and Technology*, volume 1, pages 495–498, Budapest, Hungary, 1999.

[47] D. Marcu. Towards a unified approach to memory- and statistical-based machine translation. In *Proceedings of the 39th Meeting of the Association for Computational Linguistics (ACL)*, pages 378–385, Toulouse, France, 2001.

[48] D. Marcu and W. Wong. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 133–139, Philadelphia, PA 2002.

[49] D. R. H. Miller, T. Leek, and R. M. Schwartz. BBN at TREC-7: Using hidden Markov models for information retrieval. In *Proceedings of the 7th Text REtrieval Conference*, pages 133–142, Gaithersburg, MD, 1998.

[50] A. M. Mood, F. A. Graybill, and D. C. Boes. *Introduction to the Theory of Statistics*. McGraw-Hill, Singapore, 1974.

[51] H. Ney. Speech translation: Coupling of recognition and translation. In *Proceedings of the International Conference on Acoustic, Speech and Signal Processing*, pages 517–520, Phoenix, AR, 1999.

[52] H. Ney, U. Essen, and R. Kneser. On structuring probabilistic dependences in stochastic language modelling. *Computer Speech and Language*, 8:1–38, 1994.

[53] K. Ng. A maximum likelihood ratio information retrieval model. In *Proceedings of the 8th Text REtrieval Conference*, Gaithersburg, MD, 1999.

[54] J.-Y. Nie, M. Simard, P. Isabelle, and R. Durand. Cross-language information retrieval based on parallel texts and automatic mining of parallel texts from the web. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 74–81. ACM Press, 1999.

[55] S. Niessen, F. J. Och, G. Leusch, and H. Ney. An evaluation tool for machine translation: Fast evaluation for mt research. In *Proceedings of the Second International Conference on Language Resources and Evaluation*, pages 39–45, Athens, Greece, 2000.

[56] N. J. Nilsson. *Principles of Artificial Intelligence*. Springer Verlag, Berlin, Germany, 1982.

[57] S. Nirenburg. *Machine Translation*. Cambridge University Press, 1987.

[58] F. Och and H. Ney. Discriminative training and maximum entropy models for statistical machine translation. In *ACL02: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 295–302, PA, Philadelphia, 2002.

[59] F. J. Och and H. Ney. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447, Hong-Kong, China, 2000.

[60] F. J. Och and H. Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

[61] F. J. Och, C. Tillmann, and H. Ney. Improved alignment models for statistical machine translation. In *Proceedings of the 1999 Joint SIGDAT*

Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pages 20–28, University of Maryland, MD, 1999.

[62] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. Research Report RC22176, IBM Research Division, Thomas J. Watson Research Center, 2001.

[63] K. A. Papineni, S. Roukos, and R. T. Ward. Maximum likelihood and discriminative training of direct translation models. In *Proceedings of the International Conference on Acoustic, Speech, and Signal Processing*, pages 189–192, Seattle, WA, May 1998.

[64] A. Pirkola. The effect of query structure and dictionary setups in dictionary-based cross-language information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 55–63, Melbourne, Australia, 1998.

[65] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[66] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992.

[67] V. H. Quan. *Applications of Word Graphs in Spoken Language Processing*. PhD thesis, International Doctorate School in Information and Communication Technologies, University of Trento, 2005.

[68] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In A. Weibel and K. Lee, editors, *Readings in Speech Recognition*, pages 267–296. Morgan Kaufmann, Los Altos, CA, 1990.

[69] P. Resnik, D. Oard, and G. Levow. Improved cross-language information retrieval using backoff translation. In *Proceedings of the First International Conference on Huamn Language Technology Research*, pages 117–121, San Diego, California, 2001.

[70] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *Proceedings of the 3rd Text REtrieval Conference*, pages 109–126, Gaithersburg, MD, 1994.

[71] S. Saleem, S.-C. Jou, S. Vogel, and T. Schultz. Using word lattice information for a tighter coupling in speech translation system. In *Proceedings of the 8th International Conference on Spoken Language Processing*, pages 41–44, Jeju Island, Korea, 2004.

[72] J. Savoy. Report on clef-2001 experiments: Effective combined query-translation approach. In C. Peters, M. Braschler, J. Gonzales, and M. Kluck, editors, *Evaluation of Cross-Language Information Retrieval Systems*, volume 2406 of *Lecture Notes in Computer Science*, pages 27–43, Heidelberg, Germany, 2002. Springer Verlag.

[73] C. E. Shannon. The Mathematical Theory of Communication. *Bell System Technical Journal*, 27:379–423 and 623–656, July and October 1948.

[74] F. K. Soong and E. F. Huang. A tree-trellis based fast search for finding the n-best sentence hypotheses in continuos speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 705–708, Toronto, Canada, 1991.

[75] B. Suhm, P. Geutner, T. Kemp, A. Lavie, L. Mayfield, A. McNair, I. Rogina, T. Schultz, T. Sloboda, W. Ward, M. Woszczyna, and A. Waibel. Janus: Towards multilingual spoken language translation. In *Proceedings of the ARPA Speech Spoken Language Technology Workshop*, Austin, TX, 1995.

[76] E. Sumita and H. Iida. Experiments and prospects of example-based machine translation. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pages 185–192, University of Berkeley, CA, 1991.

[77] T. Takezawa, E. Sumita, F. Sugaya, H. Yamamoto, and S. Yamamoto. Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world. In *Proceedings of 3rd International Conference on Language Resources and Evaluation (LREC)*, pages 147–152, Las Palmas, Spain, 2002.

[78] C. Tillmann. A projection extension algorithm for statistical machine translation. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, pages 1–8, Sapporo, Japan, 2003.

[79] C. Tillmann and H. Ney. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational Linguistics*, 29(1):97–133, 2003.

[80] C. Tillmann, S. Vogel, H. Ney, and A. Zubiaga. A DP-based search using monotone alignments in statistical translation. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 289–296, Somerset, NJ, 1997.

[81] B. H. Tran, F. Seide, and V. Steinbiss. A word graph based n-best search in continuous speech recognition. In *Proceedings of the International Conference of Spoken Language Processing*, Philadelphia, PA, 1996.

[82] N. Ueffing, F. Och, and H. Ney. Generation of word graphs in statistical machine translation. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing*, pages 156–163, Philadelphia, PA, 2002.

[83] E. Vidal. Finite-state speech-to-speech transaltion. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 111–114, 1997.

[84] S. Vogel, Y. Zhang, F. Huang, A. Venugopal, B. Zhao, A. Tribble, M. Eck, and A. Waibel. The CMU statistical machine translation system. In *Proceedings of the Machine Translation Summit IX*, New Orleans, LA, 2003.

[85] Y.-Y. Wang and A. Waibel. Decoding algorithm in statistical machine translation. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 366–372, Somerset, NJ, 1997.

[86] W. Weaver. Translation. In W. N. Locke and A. D. Booth, editors, *Machine Translation of languages: fourteen essays*, pages 15–23. MIT Press, Cambridge, MA, 1955.

[87] I. H. Witten and T. C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Trans. Inform. Theory*, IT-37(4):1085–1094, 1991.

[88] D. Wu. A polynomial-time algorithm for statistical machine translation. In *Proceedings of the 34th Annual Conference of the Association for Computational Linguistics*, pages 152–158, Santa Cruz, CA, 1996.

[89] J. Xu, R. Weischedel, and C. Nguyen. Evaluating a probabilistic model for cross-lingual information retrieval. In *Proceedings of the 24h Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 105–110, New Orleans, LA, 2001.

[90] K. Yamada and K. Knight. A syntactic-based statistical translation model. In *Proceedings of the 39th Meeting of the Association for Computational Linguistics (ACL)*, pages 523–530, Toulouse, France, 2001.

[91] R. Zens, F. J. Och, and H. Ney. Phrase-based statistical machine translation. In *Proceedings of the 25th Annual German Conference on Artificial Intelligence, KI-2002*, volume 2479 of *Lecture Notes in Artificial Intelligence*, pages 18–32. Springer Verlag, 2002.

[92] R. Zhang, G. Kikui, H. Yamamoto, T. Watanabe, F. Soong, and W. K. Lo. A unified approach in speech-to-speech translation: Integrating features of speech recognition and machine translation. In *Proceedings of International Conference on Computational Linguistics*, pages 1161–1167, Geneva, Switzerland, 2004.

[93] Y. Zhang and S. Vogel. Measuring confidence intervals for the machine translation evaluation metrics. In *Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 4–6, Baltimore, MD, 2004.